Introduction to Generalized Linear Models (GLMs)

Prof. Nicholas Zabaras
Center for Informatics and Computational Science
https://cics.nd.edu/
University of Notre Dame
Notre Dame, Indiana, USA

Email: nzabaras@gmail.com
URL: https://www.zabaras.com/

February 19, 2019

Contents

Following closely Chapter 9, of K. Murphy, Machine Learning: A probabilistic Perspective

Exponential Family

- Large family of useful distributions with common properties
  - Bernoulli, beta, binomial, chi-square, Dirichlet, gamma, Gaussian, geometric, multinomial, Poisson, Weibull, . .
- Not in the family: Uniform, Student's $\mathcal{T}$, Cauchy, Laplace, mixture of Gaussians, . . .
- Variable can be discrete/continuous (or vectors thereof).
- We will *focus on the conditional setting in which we have a directed model $X \rightarrow Y$ with both X & Y observed, and with $p(Y|X)$ being an exponential family distribution parametrized using Generalized Linear Models (GLIM's).*

Exponential Family

- The exponential family of distributions over $\boldsymbol{x}$, given parameters $\boldsymbol{\eta}$, is defined to be the set of distributions of the form

$$p(\boldsymbol{x}\,|\,\boldsymbol{\eta}) = h(\boldsymbol{x})g(\boldsymbol{\eta})\exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x})\right\} \; or$$

$$p(\boldsymbol{x}\,|\,\boldsymbol{\eta}) = h(\boldsymbol{x})\exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x}) - A(\boldsymbol{\eta})\right\}, \; where: A(\boldsymbol{\eta}) = -\log g(\boldsymbol{\eta})$$

$\boldsymbol{x}$ is scalar/vector, discrete/continuous. ***$\boldsymbol{\eta}$ are the natural parameters and $\boldsymbol{u(x)}$ is referred to as a sufficient statistic.***

- $g(\boldsymbol{\eta})$ *ensures that the distribution is normalized* and satisfies

$$g(\boldsymbol{\eta})\int h(\boldsymbol{x})\exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x})\right\}d\boldsymbol{x} = 1$$

- The normalization factor Z and the log of it A are defined as:

$$Z(\boldsymbol{\eta}) = \frac{1}{g(\boldsymbol{\eta})}, \; A(\boldsymbol{\eta}) = \ln Z(\boldsymbol{\eta}) = -\ln g(\boldsymbol{\eta}) = \ln \int h(\boldsymbol{x})\exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x})\right\}d\boldsymbol{x}$$

$$p(\boldsymbol{x}\,|\,\boldsymbol{\eta}) = h(\boldsymbol{x})\exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x})\right\}/Z(\boldsymbol{\eta})$$

- The space of η for which $\int h(\boldsymbol{x})\exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x})\right\}d\boldsymbol{x} < \infty$ is the ***natural parameter space***.

Canonical or Natural Parameters

- When the parameter $\boldsymbol{\theta}$ enters the exponential family as $\boldsymbol{\eta}(\boldsymbol{\theta})$, we write the probability density of the exponential family as follows:

$$p(\mathbf{x}\,|\,\boldsymbol{\theta}) = h(\mathbf{x})g\left(\boldsymbol{\eta}\left(\boldsymbol{\theta}\right)\right)\exp\left\{\boldsymbol{\eta}^T\left(\boldsymbol{\theta}\right)u(\mathbf{x})\right\}\ or$$

$$p(\mathbf{x}\,|\,\boldsymbol{\theta}) = h(\mathbf{x})\exp\left\{\boldsymbol{\eta}^T\left(\boldsymbol{\theta}\right)u(\mathbf{x}) - A\left(\boldsymbol{\eta}\left(\boldsymbol{\theta}\right)\right)\right\},$$

$$where: A\left(\boldsymbol{\eta}\left(\boldsymbol{\theta}\right)\right) = -\log g\left(\boldsymbol{\eta}\left(\boldsymbol{\theta}\right)\right)$$

- *$\boldsymbol{\eta}(\boldsymbol{\theta})$ are the canonical or natural parameters and*
- *$\boldsymbol{\theta}$ is the parameter vector of some distribution that can be written in the exponential family format.*

Joint Probability Distribution on Discrete RVs

- Any joint probability distribution on discrete random variables lies on the exponential family.* Indeed:

$$p(\boldsymbol{x}\,|\,\Psi)=\frac{1}{Z(\Psi)}\prod_{C\in\mathcal{C}}\Psi_C(\boldsymbol{x}_C)=\exp\left(\sum_{C\in\mathcal{C}}\log\Psi_C(\boldsymbol{x}_C)-\log Z(\Psi)\right)$$

$$\text{But for discrete rv's}:\Psi_C(\boldsymbol{x}_C)=\prod_{v_1^{i_1},v_2^{i_2},\ldots,v_k^{i_k}}\Psi_C\left(v_1^{i_1},v_2^{i_2},\ldots,v_k^{i_k}\right)^{\delta\left(x_1,v_1^{i_1}\right)\delta\left(x_2,v_2^{i_2}\right)\ldots\delta\left(x_k,v_k^{i_k}\right)}$$

- Substitution to the 1st Eq. gives:

$$p(\boldsymbol{x}\,|\,\Psi)=\frac{1}{Z(\Psi)}\prod_{C\in\mathcal{C}}\Psi_C(\boldsymbol{x}_C)=\exp\left(\sum_{C\in\mathcal{C}}\sum_{v_1^{i_1},v_2^{i_2},\ldots,v_k^{i_k}}\log\Psi_C\left(v_1^{i_1},v_2^{i_2},\ldots,v_k^{i_k}\right)\delta\left(x_1,v_1^{i_1}\right)\delta\left(x_2,v_2^{i_2}\right)\ldots\delta\left(x_k,v_k^{i_k}\right)-\log Z(\Psi)\right)\Rightarrow$$

$$p(\boldsymbol{x}\,|\,\Psi)=\frac{1}{Z(\Psi)}\prod_{C\in\mathcal{C}}\Psi_C(\boldsymbol{x}_C)=\exp\left(\sum_{v_1^{i_1},v_2^{i_2},\ldots,v_k^{i_k}}\sum_{C\in\mathcal{C}}\log\Psi_C\left(v_1^{i_1},v_2^{i_2},\ldots,v_k^{i_k}\right)\delta\left(x_1,v_1^{i_1}\right)\delta\left(x_2,v_2^{i_2}\right)\ldots\delta\left(x_k,v_k^{i_k}\right)-\log Z(\Psi)\right)$$

- This is in the exponential family with $\log\Psi_C\left(v_1^{i_1},v_2^{i_2},\ldots,v_k^{i_k}\right)$ corresponding to each component of $\boldsymbol{\eta}$, and $\delta\left(x_1,v_1^{i_1}\right)\delta\left(x_1,v_2^{i_2}\right)\ldots\delta\left(x_k,v_k^{i_k}\right)$ corresponding to components of the sufficient statistic $\boldsymbol{u}(\boldsymbol{x})$.

* We consider here the joint distribution of $\boldsymbol{x}$ written in terms of potentials Ψ's. In the course on probabilistic graphical models, we will see that this representation arises for rv's defined in undirected graphs where the potentials are defined over the random variables in each maximal clique c.

Exponential Family: The Bernoulli Distribution

- Consider the Bernoulli distribution considered earlier:

$$p(x \mid \mu) = \mathcal{Bern}(x \mid \mu) = \mu^x (1-\mu)^{1-x} = \exp\left\{x \ln \mu + (1-x)\ln(1-\mu)\right\} =$$

$$= \underbrace{(1-\mu)}_{g(\eta)} \exp\left\{ \underbrace{\ln\left(\frac{\mu}{1-\mu}\right)}_{\eta} x \right\}$$

$$p(\mathbf{x} \mid \boldsymbol{\eta}) = h(\mathbf{x}) g(\boldsymbol{\eta}) \exp\left\{\boldsymbol{\eta}^T u(\mathbf{x})\right\}$$
$$= h(\mathbf{x}) \exp\left\{\boldsymbol{\eta}^T u(\mathbf{x}) - A(\boldsymbol{\eta})\right\}$$

- From this we see that (note that *the relation $\mu(\eta)$ is invertible*)

Log-odds ratio

$$\eta = \ln\left(\frac{\mu}{1-\mu}\right) \Rightarrow$$

$$\mu = \sigma(\eta) = \frac{1}{1+e^{-\eta}}$$

Logistic sigmoid function

and

$$g(\eta) = 1-\mu = 1-\sigma(\eta) = \sigma(-\eta)$$

- *Finally:*

$$p(x \mid \eta) = g(\eta)\exp\{\eta x\}, \; u(x) = x = \mathbb{I}(x=1), \; h(x) = 1, \; g(\eta) = \sigma(-\eta),$$
$$A(\eta) = -\log(1-\mu) = \log(1+e^{\eta})$$

Exponential Family: The Poisson Distribution

- Consider the Poisson distribution with parameter λ:

$$p(x \mid \lambda) = \textit{Poisson}(x \mid \lambda) = \frac{\lambda^x e^{-\lambda}}{x!} = \underbrace{\frac{1}{x!}}_{h(x)} \exp\left\{ \underbrace{x}_{u(x)} \underbrace{\ln \lambda}_{\eta} - \underbrace{\lambda}_{A(\eta)} \right\}$$

- Recall that λ is the mean of the distribution and observe once more that *the relation $\lambda(\eta)$ is invertible*:

$$\eta = \ln(\lambda) \Rightarrow \lambda = e^{\eta}$$

Exponential Family: The Multinoulli Distribution

- Consider the Multinomial distribution:

$$p(\boldsymbol{x}\,|\,\boldsymbol{\mu}) = \prod_{k=1}^{M} \mu_k^{x_k} = \exp\left\{\sum_{k=1}^{M} x_k \ln \mu_k\right\} = \exp\left(\boldsymbol{\eta}^T \boldsymbol{x}\right),$$

$$\boldsymbol{x} = \left\{x_1, \ldots, x_M\right\}^T, \boldsymbol{\eta} = \left\{\eta_1, \ldots, \eta_M\right\}^T, \eta_k = \ln \mu_k$$

$$p(\boldsymbol{x}\,|\,\boldsymbol{\eta}) = h(\boldsymbol{x}) g(\boldsymbol{\eta}) \exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x})\right\}$$
$$= h(\boldsymbol{x}) \exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x}) - A(\boldsymbol{\eta})\right\}$$

- From this expression we see that $h(\boldsymbol{x}) = 1$, $u(\boldsymbol{x}) = \boldsymbol{x}$, $g(\boldsymbol{\eta}) = 1$. It appears also that $A(\eta) = 0$!

- We can resolve this problem by accounting for the dependence of μ_k, i.e. $\sum_{k=1}^{M} \mu_k = 1.$

Exponential Family: The Multinoulli Distribution

- We will express the distribution in terms of $\mu_k, k = 1, \ldots, M - 1$ subject to:

$$0 \leq \mu_k \leq 1, \quad \sum_{k=1}^{M-1} \mu_k \leq 1$$

- The multinomial distribution becomes:

$$\exp\left\{\sum_{k=1}^{M} x_k \ln \mu_k\right\} = \exp\left\{\sum_{k=1}^{M-1} x_k \ln \mu_k + \left(1 - \sum_{k=1}^{M-1} x_k\right) \ln\left(1 - \sum_{k=1}^{M-1} \mu_k\right)\right\} =$$

$$\exp\left\{\sum_{k=1}^{M-1} x_k \underbrace{\ln \frac{\mu_k}{1 - \sum_{k=1}^{M-1} \mu_k}}_{\eta_k} + \ln\left(1 - \sum_{k=1}^{M-1} \mu_k\right)\right\}$$

Exponential Family: The Multinoulli Distribution

- We identify

$$\eta_k = \ln \frac{\mu_k}{1 - \sum_{k=1}^{M-1} \mu_k} = \ln \frac{\mu_k}{\mu_M}, k = 1, \ldots, M-1$$

- Can also define:

$$\eta_M = \ln \frac{\mu_M}{\mu_M} = 0$$

- This equation can be inverted as:

$$\exp(\eta_k) = \frac{\mu_k}{1 - \sum_{k=1}^{M-1} \mu_k} \Rightarrow \sum_{k=1}^{M-1} \exp(\eta_k) = \frac{\sum_{k=1}^{M-1} \mu_k}{1 - \sum_{k=1}^{M-1} \mu_k} \Rightarrow$$

$$1 + \sum_{k=1}^{M-1} \exp(\eta_k) = \frac{1}{1 - \sum_{k=1}^{M-1} \mu_k} \Rightarrow \sum_{k=1}^{M-1} \mu_k = \frac{\sum_{k=1}^{M-1} \exp(\eta_k)}{1 + \sum_{k} \exp(\eta_k)} \Rightarrow 1 - \sum_{k=1}^{M-1} \mu_k = \left(1 + \sum_{k} \exp(\eta_k)\right)^{-1}$$

- Substitution intro the expression on the top of the slide:

$$\eta_k = \ln \frac{\mu_k}{1 - \sum_{k=1}^{M-1} \mu_k} = \ln \left[\mu_k \left(1 + \sum_{k=1}^{M-1} \exp(\eta_k) \right) \right] \Rightarrow \mu_k = \frac{\exp(\eta_k)}{1 + \sum_{k=1}^{M-1} \exp(\eta_k)}$$

Exponential Family: The Multinoulli Distribution

- This is the so called softmax function (note again *the relation $\boldsymbol{\mu}(\boldsymbol{\eta})$ is invertible)*:

$$\mu_k = \frac{\exp(\eta_k)}{1+\sum_{k=1}^{M-1}\exp(\eta_k)}$$

Softmax function

- In this reduced representation, the distribution takes the form:

$$p(\boldsymbol{x}\,|\,\boldsymbol{\eta}) = \exp\left\{\sum_{k=1}^{M-1} x_k\eta_k + \ln\left(1-\sum_{k=1}^{M-1}\mu_k\right)\right\} = \left(1+\sum_{k=1}^{M-1}\exp(\eta_k)\right)^{-1}\exp(\boldsymbol{\eta}^T\boldsymbol{x})$$

- Comparing with the generic form of the exponential family:

$$\boldsymbol{\eta} = (\eta_1,\ldots,\eta_{M-1},0)^T,\ u(\boldsymbol{x}) = \boldsymbol{x},\ h(\boldsymbol{x}) = 1,\ g(\boldsymbol{\eta}) = \left(1+\sum_{k=1}^{M-1}\exp(\eta_k)\right)^{-1}$$

$$A = -\ln g(\boldsymbol{\eta}) = \ln\left(1+\sum_{k=1}^{M-1}\exp(\eta_k)\right) = \ln\left(\sum_{k=1}^{M}\exp(\eta_k)\right)$$

Exponential Family: The Beta Distribution

- Consider the Beta distribution

$$\mathcal{Beta}(\mu \mid a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\mu^{a-1}(1-\mu)^{b-1} = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\exp\left[(a-1)\ln\mu + (b-1)\ln(1-\mu)\right]$$

- Comparing this with our exponential family:

$$p(\boldsymbol{x} \mid \boldsymbol{\eta}) = h(\boldsymbol{x})g(\boldsymbol{\eta})\exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x})\right\}$$
$$= h(\boldsymbol{x})\exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x}) - A(\boldsymbol{\eta})\right\}$$

we can easily indentify:

$$u(\mu) = (\ln\mu, \ln(1-\mu))^T,\ \boldsymbol{\eta} = (a-1, b-1)^T,\ h(\mu) = 1,\ g(a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)},$$

$$A(a,b) = \ln\frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$$

Exponential Family: Gamma Distribution

- Consider the Gamma distribution

$$\mathcal{Gamma}(\lambda \mid a,b) = \frac{b^a}{\Gamma(a)}\lambda^{a-1}e^{-b\lambda} = \frac{b^a}{\Gamma(a)}\exp\left[(a-1)\ln\lambda - b\lambda\right]$$

- Comparing this with our exponential family:

$$p(\boldsymbol{x}\mid\boldsymbol{\eta}) = h(\boldsymbol{x})g(\boldsymbol{\eta})\exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x})\right\}$$
$$= h(\boldsymbol{x})\exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x}) - A(\boldsymbol{\eta})\right\}$$

we can easily indentify:

$$u(\lambda) = (\lambda, \ln\lambda)^T,\ \boldsymbol{\eta} = (-b, a-1)^T,\ h(\lambda) = 1,\ g(a,b) = \frac{b^a}{\Gamma(a)},\ A(a,b) = \ln\frac{\Gamma(a)}{b^a}$$

Exponential Family: The Gaussian

- Consider the univariate Gaussian

$$p(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}x^2 - \frac{1}{2\sigma^2}\mu^2 + \frac{\mu}{\sigma^2}x\right\}$$

- Comparing this with our exponential family:

$$p(\boldsymbol{x} \mid \boldsymbol{\eta}) = h(\boldsymbol{x}) g(\boldsymbol{\eta}) \exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x})\right\} = h(\boldsymbol{x}) \exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x}) - A(\boldsymbol{\eta})\right\}$$

we can identify (this is a two parameter distribution):

$$u(x) = (x, x^2)^T, \boldsymbol{\eta} = \left(\frac{\mu}{\sigma^2}, -\frac{1}{2\sigma^2}\right)^T, h(x) = \frac{1}{\sqrt{2\pi}}, g(\boldsymbol{\eta}) = \left(-2\eta_2\right)^{1/2} \exp\frac{\eta_1^2}{4\eta_2}$$

$$A(\boldsymbol{\eta}) = -\frac{1}{2}\ln\left(-2\eta_2\right) - \frac{\eta_1^2}{4\eta_2}$$

Exponential Family: von Mises Distribution

- Consider the von Mises distribution

$$p(\theta \mid \theta_0, m) = \frac{1}{2\pi I_0(m)} \exp\left(m\cos(\theta - \theta_0)\right) = \frac{1}{2\pi I_0(m)} \exp\left(m\cos\theta\cos\theta_0 + m\sin\theta\sin\theta_0\right)$$

- Comparing this with our exponential family:

$$p(\boldsymbol{x} \mid \boldsymbol{\eta}) = h(\boldsymbol{x}) g(\boldsymbol{\eta}) \exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x})\right\} = h(\boldsymbol{x}) \exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x}) - A(\boldsymbol{\eta})\right\}$$

we can easily identify that:

$$u(\theta) = (\cos\theta, \sin\theta)^T, \boldsymbol{\eta} = (m\cos\theta_0, m\sin\theta_0)^T, h(\theta) = 1, g(m, \theta_0) = \frac{1}{2\pi I_0(m)},$$

$$A(m, \theta_0) = \ln\left(2\pi I_0(m)\right)$$

The Multivariate Gaussian

- The exponent in the multivariate Gaussian is:

$$-\frac{1}{2}tr\left(\Lambda \boldsymbol{x}\boldsymbol{x}^T\right)+\boldsymbol{\mu}^T\Lambda\boldsymbol{x}-\frac{1}{2}\boldsymbol{\mu}^T\Lambda\boldsymbol{\mu},\ where:\ \Lambda=\boldsymbol{\Sigma}^{-1}$$

- We need to put this in the form $p(\boldsymbol{x}\,|\,\boldsymbol{\eta})=h(\boldsymbol{x})g(\boldsymbol{\eta})\exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x})\right\}$
- The 3nd term contributes to $g(\eta)$ whereas the 2nd term is directly an inner product between $\boldsymbol{x}$ and $\boldsymbol{\xi}=\boldsymbol{L\mu}$.
- For the 1st term, define two D^2 –dimensional vectors $vec(\boldsymbol{\Lambda})$ and $vec(\boldsymbol{x}\boldsymbol{x}^T)$ that consist of the columns of $\boldsymbol{\Lambda}$ and $\boldsymbol{x}\boldsymbol{x}^T$, respectively. Then the 1st term has the form of an inner product between these two vectors. In summary:

$$\boldsymbol{\eta}=\begin{pmatrix}\boldsymbol{\xi}\\ vec(\boldsymbol{\Lambda})\end{pmatrix},\ u(\boldsymbol{x})=\begin{pmatrix}\boldsymbol{x}\\ -\frac{1}{2}vec(\boldsymbol{x}\boldsymbol{x}^T)\end{pmatrix},\ g(\boldsymbol{\eta})=\left|\Lambda\right|^{1/2}\exp\left(-\frac{1}{2}\boldsymbol{\xi}^T\Lambda^{-1}\boldsymbol{\xi}\right),\ h(\boldsymbol{x})=(2\pi)^{-D/2},\ \boldsymbol{\xi}=\boldsymbol{\Lambda\mu}$$

$$A=-\ln g(\boldsymbol{\eta})=-\frac{1}{2}\ln\left|\Lambda\right|+\frac{1}{2}\boldsymbol{\xi}^T\Lambda^{-1}\boldsymbol{\xi}$$

Computing Moments of Sufficient Statistics $u(x)$

- Differentiate wrt η the $\int p(\boldsymbol{x}|\boldsymbol{\eta})d\boldsymbol{x}=1$ for the exponential family:

$$p(\boldsymbol{x}|\boldsymbol{\eta})=h(\boldsymbol{x})g(\boldsymbol{\eta})\exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x})\right\}$$

$$\nabla g(\boldsymbol{\eta})\int h(\boldsymbol{x})\exp\left\{\boldsymbol{\eta}^T\boldsymbol{u}(\boldsymbol{x})\right\}d\boldsymbol{x}+g(\boldsymbol{\eta})\int h(\boldsymbol{x})\exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x})\right\}\boldsymbol{u}(\boldsymbol{x})d\boldsymbol{x}=0\Rightarrow$$

$$-\frac{\nabla g(\boldsymbol{\eta})}{g(\boldsymbol{\eta})}=g(\boldsymbol{\eta})\int h(\boldsymbol{x})\exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x})\right\}\boldsymbol{u}(\boldsymbol{x})d\boldsymbol{x}=\mathbb{E}\left[\boldsymbol{u}(\boldsymbol{x})\right]$$

- The above equation can be further simplified if written in terms of the partition function $Z=1/g(\eta)$ or $A=logZ$:

$$\nabla A(\boldsymbol{\eta})=\mathbb{E}\left[\boldsymbol{u}(\boldsymbol{x})\right]$$

- Let us re-write explicitly the above equation as:

$$\nabla A(\boldsymbol{\eta})=g(\boldsymbol{\eta})\int h(\boldsymbol{x})\exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x})\right\}\boldsymbol{u}(\boldsymbol{x})d\boldsymbol{x}$$

- We can compute the variance of $\boldsymbol{u}(\boldsymbol{x})$ by differentiating the Eq. above with respect to $\boldsymbol{\eta}$.

Computing Moments of Sufficient Statistics $u(x)$

$$\nabla A(\boldsymbol{\eta}) = g(\boldsymbol{\eta}) \int h(\boldsymbol{x}) \exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x})\right\} \boldsymbol{u}(\boldsymbol{x}) d\boldsymbol{x}$$

$$\nabla^2 A(\boldsymbol{\eta}) = \underbrace{\nabla g(\boldsymbol{\eta}) \int h(\boldsymbol{x}) \exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x})\right\} \boldsymbol{u}(\boldsymbol{x}) d\boldsymbol{x}}_{-\mathbb{E}[\boldsymbol{u}(\boldsymbol{x})]\mathbb{E}[\boldsymbol{u}(\boldsymbol{x})]^T} + \underbrace{g(\boldsymbol{\eta}) \int h(\boldsymbol{x}) \exp\left\{\boldsymbol{\eta}^T u(\boldsymbol{x})\right\} \boldsymbol{u}(\boldsymbol{x}) \boldsymbol{u}(\boldsymbol{x})^T d\boldsymbol{x}}_{\mathbb{E}\left[\boldsymbol{u}(\boldsymbol{x})\boldsymbol{u}(\boldsymbol{x})^T\right]}$$

where we used $\nabla A = -\frac{\nabla g}{g} = E[u(\boldsymbol{x})]$

- Thus the covariance of $u(\boldsymbol{x})$ can be expressed in terms of the 2nd derivatives of $A(\boldsymbol{\eta})$ and similarly for higher order moments.

$$\nabla^2 A(\boldsymbol{\eta}) = \mathrm{cov}\left[\boldsymbol{u}(\boldsymbol{x})\right] \textit{ so } A(\boldsymbol{\eta}) \textit{ is convex}$$

- Provided we can normalize a distribution from the exponential family, we can always find its moments by simple differentiation.

Computing Moments of Sufficient Statistics $u(x)$

$$\nabla A(\boldsymbol{\eta}) = \mathbb{E}\left[\boldsymbol{u}(\boldsymbol{x})\right]$$

$$\nabla^2 A(\boldsymbol{\eta}) = \mathrm{cov}\left[\boldsymbol{u}(\boldsymbol{x})\right] \; so \; A(\boldsymbol{\eta}) \; is \; convex$$

- Let us check these relations for the Univariate Gaussian:

$$A(\boldsymbol{\eta}) = -\frac{1}{2}\ln\left(-2\eta_2\right) - \frac{\eta_1^2}{4\eta_2}, \boldsymbol{\eta} = \left(\frac{\mu}{\sigma^2}, -\frac{1}{2\sigma^2}\right)^T, u(x) = (x, x^2)^T$$

$$\frac{\partial A(\boldsymbol{\eta})}{\partial \eta_1} = -\frac{\eta_1}{2\eta_2} = \mu = \mathbb{E}\left[X\right], \frac{\partial A(\boldsymbol{\eta})}{\partial \eta_2} = -\frac{1}{2\eta_2} + \frac{\eta_1^2}{4\eta_2^2} = \mu^2 + \sigma^2 = \mathbb{E}\left[X^2\right]$$

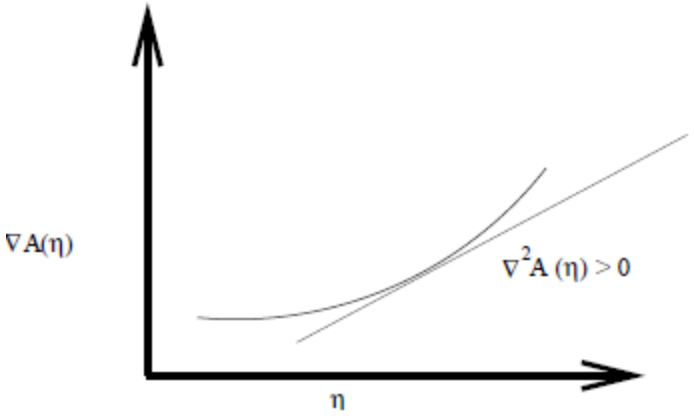
$$\frac{\partial^2 A(\boldsymbol{\eta})}{\partial \eta_1^2} = -\frac{1}{2\eta_2} = \sigma^2 = \mathrm{var}\left[X\right], etc.$$

Moment Parametrization

- We have shown that we can compute the mean of the distribution $\mu = \mathbb{E}[u(\boldsymbol{x})]$ in terms of the canonical parameter η:

$$\mu = \mathbb{E}\left[\boldsymbol{u}(\boldsymbol{x})\right] = \nabla A(\boldsymbol{\eta})$$

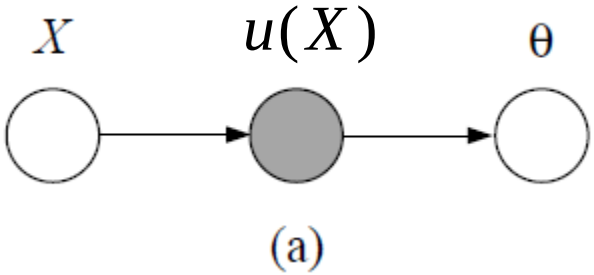
- We have also shown that $A(\eta)$ is a convex function. Since for a convex function there is one-to-one relation between the argument of the function and its derivative, the mapping $\mu(\eta)$ is invertable.



- Thus the exponential family of distributions can also be parameterized in terms of μ (*moment parametrization*) exactly as one often starts in introducing distributions for different random variables.

Sufficiency

- *$u(X)$ is sufficient for θ if there is no information in X regarding θ beyond that in $u(X)$.* Having observed $u(X)$, we can throw away X for the purposes of inference with respect to θ.
- In the Bayesian approach in the Fig shown, we treat θ as a rv and say that $u(X)$ is sufficient for θ if the following CI statement holds:



(a)

$$\theta \perp X \mid u(X)$$

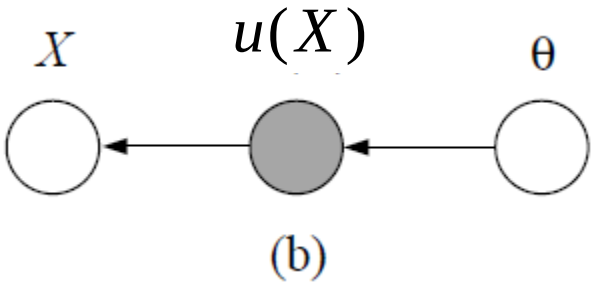
$$p\left(\theta \mid u(x), x\right) = p\left(\theta \mid u(x)\right)$$

- Thus, *$u(X)$ contains all the needed information in X about θ.*

Frequentist Definition: Sufficiency

- The model in Fig b shown asserts the same CI relations as shown in Fig a earlier but has different parametrization.

$$p\left(x \mid u(x), \theta\right) = p\left(x \mid u(x)\right)$$



(b)

- Treating θ as a label, we can see the above CI statement as a frequentist definition of sufficiency.
- *$u(X)$ is sufficient for θ if the $p(x|u(x))$ is not a function of θ.*
- The two approaches discussed imply a particular factorization of $p(x|\theta)$.

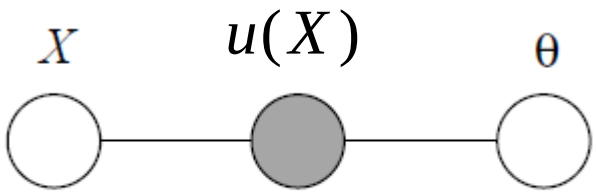
Neymann Factorization Theorem

- From the undirected graph (that expresses the same CI relations as the two earlier graphs), we can factorize as:

$$p\left(x,u(x),\theta\right)=g_1\left(u(x),\theta\right)g_2\left(x,u(x)\right)$$

- On the left, $u(x)$ is a deterministic function of x and can be dropped as an argument:

$$p\left(x,\theta\right)=g_1\left(u(x),\theta\right)g_2\left(x,u(x)\right)$$



(c)

- One can derive for given ψ_1,ψ_2:

$$p\left(x\,|\,\theta\right)=p\left(x,\theta\right)/\,p\left(\theta\right)=\psi_1\left(u(x),\theta\right)\psi_2\left(x,u(x)\right)$$

- We can now see why $u(x)$ was sufficient statistic for $\boldsymbol{\eta}$ in the exponential family:

$$p(\boldsymbol{x}\,|\,\boldsymbol{\eta})=\underbrace{h(\boldsymbol{x})}_{\psi_2(u(x),x)}\underbrace{\exp\left\{\boldsymbol{\eta}(\theta)^T u(\boldsymbol{x})-A\left(\boldsymbol{\eta}(\theta)\right)\right\}}_{\psi_1(u(x),\theta)}$$

MLE for the Exponential Family

- The joint density for a data $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ *is itself an exp. distribution with sufficient statistics* $\sum_{n=1}^{N} u(\boldsymbol{x}_n)$:

$$p(\boldsymbol{X} \mid \boldsymbol{\eta}) = \prod_{n=1}^{N} \left(h(\boldsymbol{x}_n) g(\boldsymbol{\eta}) \exp\left\{ \boldsymbol{\eta}^T u(\boldsymbol{x}_n) \right\} \right) = \prod_{n=1}^{N} \left(h(\boldsymbol{x}_n) \right) g(\boldsymbol{\eta})^N \exp\left\{ \boldsymbol{\eta}^T \sum_{n=1}^{N} u(\boldsymbol{x}_n) \right\} \Rightarrow$$

$$\ln p(\boldsymbol{X} \mid \boldsymbol{\eta}) = \sum_{n=1}^{N} h(\boldsymbol{x}_n) + N \ln g(\boldsymbol{\eta}) + \boldsymbol{\eta}^T \sum_{n=1}^{N} u(\boldsymbol{x}_n) = \sum_{n=1}^{N} h(\boldsymbol{x}_n) - NA(\boldsymbol{\eta}) + \boldsymbol{\eta}^T \sum_{n=1}^{N} u(\boldsymbol{x}_n)$$

- *The exponential family is the only family of distributions* ***with finite sufficient statistics*** *(size independent of the data set size).*
- *The log likelihood is concave (A convex) and has a unique maximum.*
- Maximizing wrt $\boldsymbol{\eta}$ gives: $\nabla A(\boldsymbol{\eta}_{ML}) = \mathbb{E}\left[\boldsymbol{u}(\boldsymbol{x})\right] = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{u}(\boldsymbol{x}_n)$
- *At the MLE, the empirical average of the sufficient statistic is equal the model's theoretical expected sufficient statistics (moment matching).*
- *Thus to find the expected value of the sufficient statistics, one can use directly the data without having to estimate* $\boldsymbol{\eta}$. When $\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{x}$, the above allows us to compute the expectation of $\boldsymbol{x}$ directly from the data.

MLE for the Exponential Family

$$\nabla A(\boldsymbol{\eta}_{ML}) = \mathbb{E}\left[\boldsymbol{u}(\boldsymbol{x})\right] = \frac{1}{N}\sum_{n=1}^{N} u(\boldsymbol{x}_n)$$

- Using the sufficient statistic, one can in principle invert the above equ. to compute $\boldsymbol{\eta}_{MLE}$. For example, for the Bernoulli distribution,

$$p(x\,|\,\eta) = g(\eta)\exp\{\eta x\},\ u(x) = \mathbb{I}(x=1),\ h(x) = 1,$$

$$\mu = \frac{1}{1+e^{-\eta}},\ g(\eta) = \frac{1}{1+e^{\eta}},\ \eta = \ln\left(\frac{\mu}{1-\mu}\right)$$

and thus:

$$\mathbb{E}[X] = p(X=1) = \bar{\mu} \equiv \mu_{MLE} = \frac{1}{N}\sum_{n=1}^{N}\mathbb{I}(x_n = 1)$$

and

$$\eta_{MLE} = \ln\left(\frac{\bar{\mu}}{1-\bar{\mu}}\right)$$

MLE and Kullback-Leibler Distance

- A useful property for the MLE (and not just a property for the exponential family of distributions) is the following:
- Minimizing the KL distance to the empirical distribution is equivalent to maximizing the likelihood.
- Indeed, let us consider the model $logp(x|\theta)$ and the empirical distribution:

$$p_{emp}(x) = \frac{1}{N}\sum_{n=1}^{N}\delta\left(x, x_n\right)$$

- We can then derive the following:

$$\sum_x p_{emp}(x)\log p(x\,|\,\theta) = \frac{1}{N}\sum_{n=1}^{N}\sum_x \delta\left(x, x_n\right)\log p(x\,|\,\theta) = \frac{1}{N}\sum_{n=1}^{N}\log p(x_n\,|\,\theta) = \frac{1}{N}\ell(\theta\,|\,\mathcal{D})$$

and from this:

$$KL\left(p_{emp}(x), p(x\,|\,\theta)\right) = \sum_x p_{emp}(x)\log\frac{p_{emp}(x)}{p(x\,|\,\theta)} = \sum_x p_{emp}(x)\log p_{emp}(x) - \sum_x p_{emp}(x)\log p(x\,|\,\theta)$$

$$= \sum_x p_{emp}(x)\log p_{emp}(x) - \frac{1}{N}\ell(\theta\,|\,\mathcal{D})$$

- Since the 1st term is independent of θ, the assertion is proved.

Conjugate Priors

- We have already encountered the concept of a conjugate prior:
  - For the Bernoulli, the conjugate prior is the Beta distribution
  - For the Gaussian, the conjugate prior for the mean is a Gaussian, and the conjugate prior for the precision is the Wishart distribution
- In general, for a given probability distribution $p(\boldsymbol{x}|\boldsymbol{\eta})$, we can seek a prior $p(\boldsymbol{\eta})$ that is conjugate to the likelihood function, so that the posterior distribution has the same functional form as the prior. For any member of the exponential family,

$$p(\boldsymbol{x}\,|\,\boldsymbol{\theta}) = h(\boldsymbol{x})g(\boldsymbol{\eta}(\boldsymbol{\theta}))\exp\left\{\boldsymbol{\eta}^T(\boldsymbol{\theta})u(\boldsymbol{x})\right\}$$

there exists a conjugate prior that can be written in the form

$$p(\boldsymbol{\theta}\,|\,\nu_0,\tau_0) \propto g\left(\boldsymbol{\eta}(\boldsymbol{\theta})\right)^{\nu_0}\exp\left\{\boldsymbol{\eta}^T(\boldsymbol{\theta})\tau_0\right\} = \exp\left\{\nu_0\boldsymbol{\eta}^T(\boldsymbol{\theta})\bar{\tau}_0 - A\left(\boldsymbol{\eta}(\boldsymbol{\theta})\right)\nu_0\right\},\ where: \tau_0 \equiv \nu_0\bar{\tau}_0$$

- In normalized form, we write:

$$p(\boldsymbol{\theta}\,|\,\nu_0,\tau_0) = \frac{1}{Z(\nu_0,\tau_0)}g\left(\boldsymbol{\eta}(\boldsymbol{\theta})\right)^{\nu_0}\exp\left\{\boldsymbol{\eta}^T(\boldsymbol{\theta})\tau_0\right\} = \frac{1}{Z(\nu_0,\tau_0)}\exp\left\{\nu_0\boldsymbol{\eta}^T(\boldsymbol{\theta})\bar{\tau}_0 - A\left(\boldsymbol{\eta}(\boldsymbol{\theta})\right)\nu_0\right\}$$

$$where: Z(\nu_0,\tau_0) = \int \exp\left\{\nu_0\boldsymbol{\eta}^T(\boldsymbol{\theta})\bar{\tau}_0 - A\left(\boldsymbol{\eta}(\boldsymbol{\theta})\right)\nu_0\right\}d\boldsymbol{\theta}$$

Conjugate Priors

$$p(\boldsymbol{X}\,|\,\boldsymbol{\theta}) = \prod_{n=1}^{N}\left(h(\boldsymbol{x}_n)g\left(\boldsymbol{\eta}(\boldsymbol{\theta})\right)\exp\left\{\boldsymbol{\eta}^T(\boldsymbol{\theta})u(\boldsymbol{x}_n)\right\}\right) = \prod_{n=1}^{N}\left(h(\boldsymbol{x}_n)\right)g\left(\boldsymbol{\eta}(\boldsymbol{\theta})\right)^N \exp\left\{\boldsymbol{\eta}^T(\boldsymbol{\theta})\sum_{n=1}^{N}u(\boldsymbol{x}_n)\right\}$$

$$p(\boldsymbol{\theta}\,|\,\nu_0,\tau_0) = \frac{1}{Z(\nu_0,\tau_0)}g\left(\boldsymbol{\eta}(\boldsymbol{\theta})\right)^{\nu_0}\exp\left\{\boldsymbol{\eta}^T(\boldsymbol{\theta})\tau_0\right\} = \frac{1}{Z(\nu_0,\tau_0)}\exp\left\{\nu_0\boldsymbol{\eta}^T(\boldsymbol{\theta})\bar{\tau}_0 - A\left(\boldsymbol{\eta}(\boldsymbol{\theta})\right)\nu_0\right\}$$

- Using $\bar{\boldsymbol{u}} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{u}(\boldsymbol{x}_n)$, the posterior becomes (this form justifies $\bar{\tau}_0$):

$$p(\boldsymbol{\theta}\,|\,\boldsymbol{X},\boldsymbol{\chi},\nu) \propto g\left(\boldsymbol{\eta}(\boldsymbol{\theta})\right)^{\nu_0+N}\exp\left\{\boldsymbol{\eta}^T(\boldsymbol{\theta})\left(\sum_{n=1}^{N}\boldsymbol{u}(\boldsymbol{x}_n)+\nu\bar{\tau}_0\right)\right\} = g\left(\boldsymbol{\eta}(\boldsymbol{\theta})\right)^{\nu_0+N}\exp\left\{\boldsymbol{\eta}^T(\boldsymbol{\theta})\left(N\bar{\boldsymbol{u}}+\nu_0\bar{\tau}_0\right)\right\}$$

- The parameter ν_0 can be interpreted as *effective number of fictitious observations* in the prior each of which has a value for the sufficient statistic equal to $\bar{\tau}_0$.

$$p\left(\boldsymbol{\theta}\,|\,\boldsymbol{X},\nu_N,\tau_N\right) = \frac{1}{Z(\nu_N,\tau_N)}g\left(\boldsymbol{\eta}(\boldsymbol{\theta})\right)^{\nu_N}\exp\left\{\left(N+\nu_0\right)\boldsymbol{\eta}^T(\boldsymbol{\theta})\frac{N\bar{\boldsymbol{u}}+\nu_0\bar{\tau}_0}{N+\nu_0}\right\} = \frac{1}{Z(\nu_N,\tau_N)}g\left(\boldsymbol{\eta}(\boldsymbol{\theta})\right)^{\nu_N}\exp\left\{\nu_N\boldsymbol{\eta}^T(\boldsymbol{\theta})\bar{\tau}_N\right\},$$

$$\text{where } \nu_N = \nu_0 + N,\ \bar{\tau}_N = \frac{N\bar{\boldsymbol{u}}+\nu_0\bar{\tau}_0}{N+\nu_0},\ \tau_N = \nu_N\bar{\tau}_N = N\bar{\boldsymbol{u}}+\nu_0\bar{\tau}_0 = \sum_{i=1}^{N}\boldsymbol{u}(\boldsymbol{x}_i)+\tau_0$$

Posterior Predictive

- Let $\boldsymbol{u}(\boldsymbol{X}) = \sum_{i=1}^{N} \boldsymbol{u}(\boldsymbol{x}_i), \boldsymbol{u}(\boldsymbol{X}') = \sum_{i=1}^{N'} \boldsymbol{u}(\boldsymbol{x}_i'),$ the posterior predictive is then:

$$p(\boldsymbol{X}' \mid \boldsymbol{X}) = \int p(\boldsymbol{X}' \mid \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \boldsymbol{X}) d\boldsymbol{\theta}$$
$$= \prod_{i=1}^{N'} h(\boldsymbol{x}_i') \int g(\boldsymbol{\eta})^{N'} \exp\left\{\boldsymbol{\eta}^T(\boldsymbol{\theta}) \boldsymbol{u}(\boldsymbol{X}')\right\} \frac{1}{Z(\nu_0 + N, \boldsymbol{u}(\boldsymbol{X}) + \tau_0)} g\left(\boldsymbol{\eta}(\boldsymbol{\theta})\right)^{\nu_N} \exp\left\{\boldsymbol{\eta}^T(\boldsymbol{\theta})\left(\boldsymbol{u}(\boldsymbol{X}) + \tau_0\right)\right\} d\boldsymbol{\theta}$$

- This is simplified as follows:

$$p(\boldsymbol{X}' \mid \boldsymbol{X}) = \prod_{i=1}^{N'} h(\boldsymbol{x}_i') \frac{1}{Z\left(\nu_0 + N, \boldsymbol{u}(\boldsymbol{X}) + \tau_0\right)} \int g\left(\boldsymbol{\eta}(\boldsymbol{\theta})\right)^{N' + \nu_N} \exp\left\{\boldsymbol{\eta}^T(\boldsymbol{\theta})\left(\boldsymbol{u}(\boldsymbol{X}') + \boldsymbol{u}(\boldsymbol{X}) + \tau_0\right)\right\} d\boldsymbol{\theta}$$
$$= \prod_{i=1}^{N'} h(\boldsymbol{x}_i') \frac{Z\left(\nu_0 + N + N', \boldsymbol{u}(\boldsymbol{X}') + \boldsymbol{u}(\boldsymbol{X}) + \tau_0\right)}{Z\left(\nu_0 + N, \boldsymbol{u}(\boldsymbol{X}) + \tau_0\right)}$$

- If $N = 0$, this becomes the marginal likelihood of $\boldsymbol{X}'$, which reduces to the normalizer of the posterior divided by the normalizer of the prior multiplied by a constant.

Beta/Bernoulli: Posterior Predictive

- Consider a Bernoulli likelihood with a Beta prior. The likelihood takes the familiar exponential distribution form:

$$p(\mathcal{D}\,|\,\theta)=\theta^{\sum_i x_i}(1-\theta)^{N-\sum_i x_i}=(1-\theta)^N\exp\left(\left(\log\frac{\theta}{1-\theta}\right)\sum_i x_i\right)$$

- The conjugate prior is a Beta: $p(\theta\,|\,\nu_0,\tau_0)\propto(1-\theta)^{\nu_0}\exp\left(\log\left(\frac{\theta}{1-\theta}\right)\tau_0\right)=\theta^{\tau_0}(1-\theta)^{\nu_0-\tau_0}$

$$p(\theta\,|\,\nu_0,\tau_0)=\mathcal{Beta}(\alpha,\beta),\alpha=\tau_0+1,\beta=\nu_0-\tau_0+1,$$

- Thus the posterior becomes: $p(\theta\,|\,\mathcal{D})\propto\theta^{\tau_0+s}(1-\theta)^{\nu_0-\tau_0+N-s}\Rightarrow$

$$p(\theta\,|\,\mathcal{D})=\mathcal{Beta}(\alpha_N,\beta_N),\alpha_N=\alpha+s,\beta_N=\beta+(N-s),s=\sum_i\mathbb{I}(x_i=1)$$

- Let s the number of heads in the past data. The probability of $s'=\sum_{i=1}^{m}\mathbb{I}(x_i'=1)$ future heads in m trials is then:

$$p(\theta\,|\,\mathcal{D})=\int\theta^{s'}(1-\theta)^{m-s'}\frac{\Gamma(\alpha_N+\beta_N)}{\Gamma(\alpha_N)\Gamma(\beta_N)}\theta^{\alpha_N-1}(1-\theta)^{\beta_N-1}d\theta=\frac{\Gamma(\alpha_N+\beta_N)}{\Gamma(\alpha_N)\Gamma(\beta_N)}\frac{\Gamma(\alpha_{N+m})\Gamma(\beta_{N+m})}{\Gamma(\alpha_{N+m}+\beta_{N+m})}$$

$$\alpha_{N+m}=\alpha_N+s',\beta_{N+m}=\beta_N+(m-s')$$

Maximum Entropy and Exponential Family

- If nothing is known about a distribution except that it belongs to a certain class, then the distribution with the largest entropy should be chosen as the default.[a]
- The entropy is defined as
  - discrete case $$H(\pi) = -\sum_k \pi(\theta_k)\log\left(\pi(\theta_k)\right)$$
- When some statistics (moments) of the distribution are known,

$$\mathbb{E}_\pi\left[g_k(\theta)\right] = w_k, k = 1,..,K$$

the maximum entropy distribution is of the form (λ's are the Lagrange multipliers enforcing the constraints):

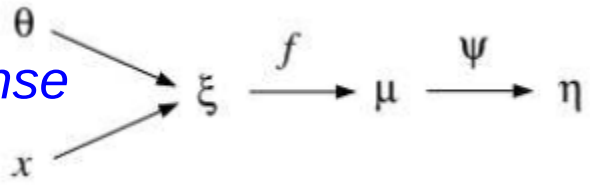
$$\pi(\theta_i) = \frac{\exp\left(-\sum_{k=1}^{K}\lambda_k g_k(\theta_i)\right)}{\sum_j \exp\left(-\sum_{k=1}^{K}\lambda_k g_k(\theta_j)\right)}, \lambda_k = Lagrange\ multipliers$$

- *Thus the MaxEnt distribution has the form of the exponential family.*

[a] C. P. Robert, *The Bayesian Choice, Springer, 2nd edition,* chapter 3 (full text available for Notre Dame students)

Generalized Linear Models

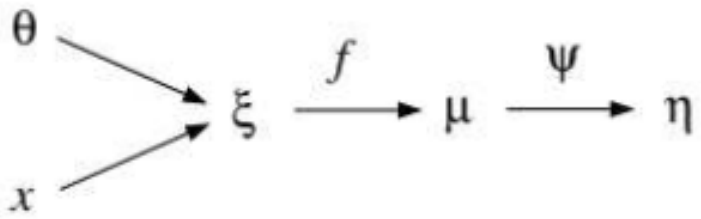
- We now study the regression between X and Y using a GLIM.
- We choose a particular conditional expectation of Y. We denote the modeled value of conditional expectation as $\mu = f(\theta^T x)$.
- For linear regression, *GLIM extends these ideas beyond the Gaussian, Bernoulli and multinomial setting to the more general exponential family.*
- x enters linearly as $\theta^T x$ and f *is called a response function.* Ψ is a one-to-one map of μ to η.



- To specify a GLIM we need (a) a choice of exponential family distribution, and (b) a choice of the response function $f()$.
- Choosing the exponential family distribution is strongly constrained by the nature of the data.
- In choosing $f(.)$, note that it needs to be both monotonic and differentiable. However*, there is a particular response function (canonical response function) that is uniquely associated with a given exponential family distribution.*

The mean and link functions

- To convert from the mean parameter to the natural parameter we use a function ψ, so $\eta = \Psi(\mu).\psi$ is uniquely determined by the form of the exponential distribution.



- This is an invertible mapping, so we have $\mu = \Psi^{-1}(\eta)$
- We know that the mean is given by the derivative of the partition function, so we have $\mu = \Psi^{-1}(\eta) = A'(\eta)$.
- In the context of regression define a linear function of the inputs: $\xi_i = \boldsymbol{\theta}^T \boldsymbol{x}_i$
- We make the mean of the distribution to be some invertible monotonic function of ξ_i . This function, known as the ***mean function***, is denoted by f.

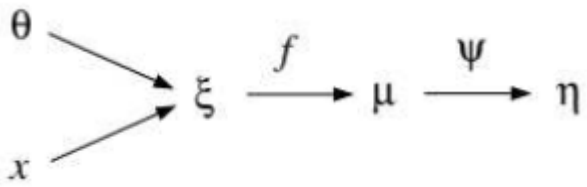
$$\mu_i = f\ \xi_i\ = f\ \boldsymbol{\theta}^T \mathbf{x}_i$$

- The inverse of the mean function, namely $f^{-1}()$, is called the ***link function***.
- In logistic regression, we set $\mu_i = f(\xi_i) = sigm(\xi_i)$.

Canonical Response Function

- Canonical response function:

$$f() = \Psi^{-1}()$$
$$\xi = \eta$$



- *If we decide to use the canonical response function, the choice of the exponential family density completely determines the GLIM.*

$$\xi = f^{-1}(\mu) = \Psi(\mu) = \eta$$

MLE & Canonical Response Function

- Consider a regression problem with data $\mathcal{D} = \{(\boldsymbol{x_i}, y_i\}, i = 1, \ldots N\}$. *The log likelihood for a GLIM is:*

$$\ell(\theta, \mathcal{D}) = \sum_{n=1}^{N} \log h(y_n) + \sum_{n=1}^{N} \underbrace{\eta_n \; y_n - A(\eta_n)}_{\psi(\mu_n)}, \; where: \; \mu_n = f(\xi_n) \; with \; \xi_n = \boldsymbol{\theta}^T \boldsymbol{x}_n$$

- *For a canonical response, this is simplified as:*

$$\ell(\theta, \mathcal{D}) = \sum_{n=1}^{N} \log h(y_n) + \boldsymbol{\theta}^T \underbrace{\sum_{n=1}^{N} \boldsymbol{x}_n y_n}_{Sufficient\ statistic\ for\ \boldsymbol{\theta}} - \sum_{n=1}^{N} A(\eta_n)$$

- Regardless of N, the size of the sufficient statistic is fixed: the dimension of $\boldsymbol{x}_n$ - important reason for using canonical response.

$$\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}, \mathcal{D}) = \sum_{n=1}^{N} (y_n - A'(\eta_n)) \nabla_{\boldsymbol{\theta}} \eta_n = \sum_{n=1}^{N} (y_n - \mu_n) \nabla_{\boldsymbol{\theta}} \eta_n = \sum_{n=1}^{N} (y_n - \mu_n) \boldsymbol{x}_n \; or \; \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}, \mathcal{D}) = \boldsymbol{X}^T (\boldsymbol{y} - \boldsymbol{\mu})$$

$\boldsymbol{x}_n$

- *This is a general expression for GLM with exponential family distributions and the canonical response function*.

Batch Algorithm

- The Hessian can now be computed from

$$\nabla_{\boldsymbol{\theta}}\ell\left(\boldsymbol{\theta},\mathcal{D}\right)=\sum_{n=1}^{N}\left(y_n-\mu_n\right)\boldsymbol{x}_n \text{ or } \nabla_{\boldsymbol{\theta}}\ell\left(\boldsymbol{\theta},\mathcal{D}\right)=\boldsymbol{X}^T\left(\boldsymbol{y}-\boldsymbol{\mu}\right)$$

as:

$$\boldsymbol{H}=\nabla^2_{\boldsymbol{\theta}}\ell\left(\boldsymbol{\theta},\mathcal{D}\right)=-\sum_{n=1}^{N}\frac{d\mu_n}{d\eta_n}\boldsymbol{x}_n\boldsymbol{x}_n^T \text{ or } \nabla_{\boldsymbol{\theta}}\ell\left(\boldsymbol{\theta},\mathcal{D}\right)=-\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}, \text{where}: \boldsymbol{W}=\left\{\frac{d\mu_1}{d\eta_1},\ldots,\frac{d\mu_n}{d\eta_n}\right\}$$

- To estimate parameters in the canonical response function choice, one can use iteratively reweighted least squares (IRLS) algorithm
- *The batch Newton algorithm now takes the familiar IRLS form:*

$$\theta^{t+1}=\theta^t+\left(\boldsymbol{X}^T\boldsymbol{W}^t\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\left(\boldsymbol{y}-\boldsymbol{\mu}^t\right)=\left(\boldsymbol{X}^T\boldsymbol{W}^t\boldsymbol{X}\right)^{-1}\left(\boldsymbol{X}^T\boldsymbol{W}^t\boldsymbol{X}\theta^t+\boldsymbol{X}^T\left(\boldsymbol{y}-\boldsymbol{\mu}^t\right)\right)$$

$$=\left(\boldsymbol{X}^T\boldsymbol{W}^t\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{W}^t\left(\underbrace{\boldsymbol{X}\theta^t+\boldsymbol{W}^{t-1}\left(\boldsymbol{y}-\boldsymbol{\mu}^t\right)}_{\eta}\right)=\left(\boldsymbol{X}^T\boldsymbol{W}^t\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{W}^t\left(\boldsymbol{\eta}+\boldsymbol{W}^{t^{-1}}\left(\boldsymbol{y}-\boldsymbol{\mu}^t\right)\right)$$

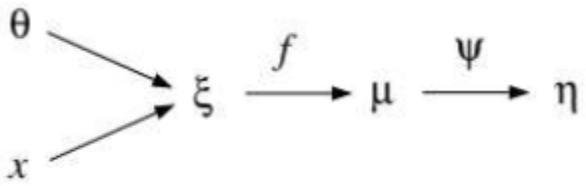
- For non-canonical response functions, the Hessian has an extra term that contains the factor $(y-\mu)$. When we take expectations this term vanishes! So using the expected Hessian in the Newton method the algorithm looks essentially the same (Fisher Scoring algorithm).

Sequential Estimation - LMS

- An on-line estimation algorithm can be obtained by following the stochastic gradient of the log likelihood function.

$$\theta^{t+1} = \theta^{t} + \rho\left(y_n - \mu_n^t\right)x_n, \mu_n^t = f\left(\theta^{t^T} x_n\right)$$

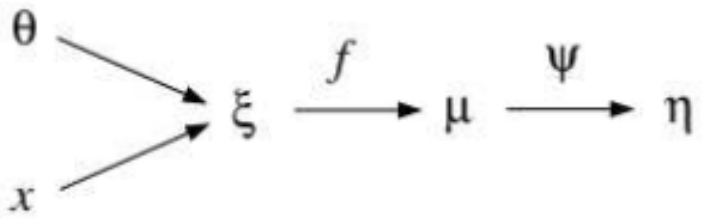
- If we do not use the canonical response function, then the gradient also includes the derivatives of $f()$ and $\Psi()$. These can be viewed as scaling coefficients that alter the step size, but otherwise leave the general LMS form intact.



- *The LMS algorithm is the generic stochastic gradient algorithm for models throughout the GLIM family.*

Canonical Link Function: Gaussian Regression

- As discussed, one particularly simple form of the link function is $f^{-1} = \psi$ *where* ψ *is defined by the exponential family*; this is called the ***canonical link function***.



- Consider this case in a regression problem for which $\mu_i = f(\xi_i),\ \xi_i = \boldsymbol{\theta}^T \boldsymbol{x}_i$ and the exponential family likelihood is as follows;

$$p\ \ y_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}, \sigma^2\ \ = \log h(y_i, \sigma^2) + \frac{\psi\ \ \mu_i}{\sigma^2} = \log h(y_i, \sigma^2) + \frac{y_i \boldsymbol{\theta}^T \boldsymbol{x}_i - A(\boldsymbol{\theta}^T \boldsymbol{x}_i)}{\sigma^2},$$

$$y_i\ \ \eta_i\ \ - A(\eta_i)$$

Note the factor σ^2 is introduced for convenience e.g. to allow us for Gaussian likelihood to use $\eta = \mu$ rather than $\eta = \mu/\sigma^2$. This would also make $\eta_i = \xi_i = \mu_i = \boldsymbol{\theta}^T \boldsymbol{x}_i$ possible. With this definition note that

and

$$\log p\left(y_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}, \sigma^2\right) = \frac{y_i \mu_i - \frac{\mu_i^2}{2}}{\sigma^2} - \frac{1}{2}\left(\frac{y_i^2}{\sigma^2} + \log\left(2\pi\sigma^2\right)\right) \Rightarrow A(\eta) = \frac{\eta^2}{2}.$$

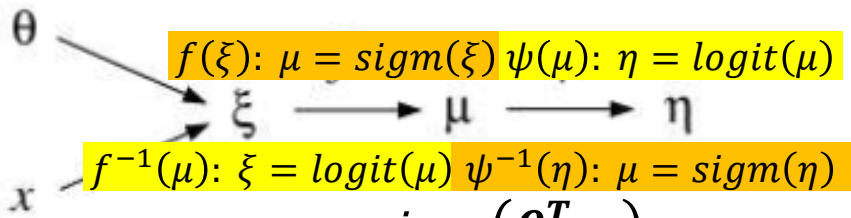
$$\mathbb{E}\ \ y_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}, \sigma^2\ \ = \mu_i = A'\ \ \eta_i\ \ ,\ \text{var}\ \ y_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}, \sigma^2\ \ = \sigma_i^2 = A''\ \ \eta_i\ \ \sigma^2$$

- *For Gaussian likelihood*, $\psi(\mu)$ and thus the link function $f^{-1}(\mu)$ are identity.

Canonical Link Function: Binomial Regression

- In GLMs the output density is in the exponential family and *the mean parameters are a linear combination of the inputs* passed through a possibly nonlinear function. We consider here binomial regression with $y_i \in 0,1,2,\ldots,N_i$. We take $\sigma^2 = 1$

$$p\ y_i \mid \eta, \sigma^2 = \exp\left[y_i \eta - A(\eta) + c\ y_i\right]$$



- The binomial likelihood is as follows:

$$\log p\ y_i \mid \boldsymbol{x}_i, \boldsymbol{\theta} = y_i \log \frac{\pi_i}{1-\pi_i} + N_i \log\ 1-\pi_i + \log \binom{N_i}{y_i} \Rightarrow$$

$$\eta_i = \log \frac{\pi_i}{1-\pi_i} = \boldsymbol{\theta}^T \boldsymbol{x}_i \quad \pi_i = sigm\ \eta_i \quad , A = -N_i \log\ 1-\pi_i = N_i \log\ 1+e^{\eta_i}$$

- From this we derive:

$$\mathbb{E}\ y_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}, \sigma^2 = \mu_i = A'\ \eta_i = N_i sigm(\eta_i) = N_i \pi_i ,$$

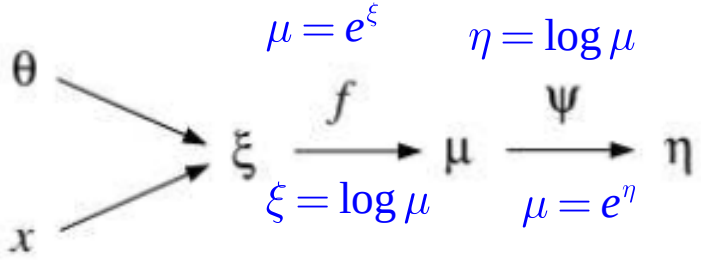
$$\text{var}\ y_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}, \sigma^2 = \sigma_i^2 = A''\ \eta_i = N_i \pi_i\ 1-\pi_i$$

- McCullagh, P. and J. Nelder (1989). *Generalized linear models*. Chapman and Hall. 2nd edition.

Canonical Link Function: Poisson Regression

- Let us now consider Poisson regression.

$$p\ \ y_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}, \sigma^2\ \ = \exp\left[\frac{y_i \boldsymbol{\theta}^T \boldsymbol{x}_i - A(\boldsymbol{\theta}^T \boldsymbol{x}_i)}{\sigma^2} + c\ \ y_i, \sigma^2\ \right]$$



- The log-likelihood takes the form:

$$\log p\ \ y_i \mid \boldsymbol{x}_i, \boldsymbol{w}\ \ = y_i \log \mu_i - \mu_i - \log\ \ y_i! \Rightarrow$$

$$\eta_i = \log \mu_i = \xi_i = \boldsymbol{\theta}^T \boldsymbol{x}_i \quad \mu_i = e^{\eta_i} \quad , A = \mu_i = e^{\eta_i}$$

- Thus we can define the mean and variance as:

$$\mathbb{E}\ \ y_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}, \sigma^2\ \ = A'\ \ \eta_i\ \ = e^{\eta_i} = \mu_i,$$

$$\text{var}\ \ y_i \mid \boldsymbol{x}_i, \boldsymbol{\theta}, \sigma^2\ \ = A''\ \ \eta_i\ \ = e^{\eta_i} = \mu_i$$

- Poisson regression is common in biostatistics where e.g. y_i may represent the number of diseases of a given person or the number of reads at a genomic location in a high-throughput sequencing context.

- Kuan, P., G. Pan, J. A. Thomson, R. Stewart, and S. Keles (2009). A hierarchical semi-Markov model for detecting enrichment with application to ChIP-Seq experiments. Technical report, U. Wisconsin.

MLE Estimation

- The log-likelihood has the form:

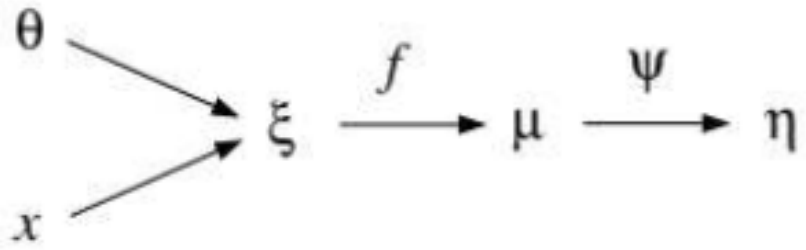
$$l\ \boldsymbol{\theta}\ = \log p\ \ \mathcal{D}\,|\,\boldsymbol{\theta}\ = \frac{1}{\sigma^2}\sum_{i=1}^{N}\ell_i,\ \ell_i = y_i\eta_i - A(\eta_i)$$

- The gradient is easily computed using the chain rule:

$$\frac{d\ell_i}{d\theta_j} = \frac{d\ell_i}{d\eta_i}\frac{d\eta_i}{d\mu_i}\frac{d\mu_i}{d\xi_i}\frac{d\xi_i}{d\theta_j} = \ y_i - A'(\eta_i)\ \frac{d\eta_i}{d\mu_i}\frac{d\mu_i}{d\xi_i}x_{ij} = \ y_i - \mu_i\ \ \frac{d\eta_i}{d\mu_i}\frac{d\mu_i}{d\xi_i}x_{ij}$$

- If we use a canonial link $\eta_i = \xi_i$:

$$\frac{d\ell_i}{d\theta_j} = \ y_i - \mu_i\ \ x_{ij}\ or\ \frac{d\ell_i}{d\boldsymbol{\theta}} = \ y_i - \mu_i\ \ \boldsymbol{x}_i$$



- Thus the log likelihood takes a familiar form that can be used directly in stochastic optimization:

$$\nabla_{\theta} l\ \boldsymbol{\theta}\ = \log p\ \ \mathcal{D}\,|\,\boldsymbol{\theta}\ = \frac{1}{\sigma^2}\sum_{i=1}^{N}\underbrace{y_i - \mu_i}_{errors}\ \boldsymbol{x}_i$$

Computing the Hessian

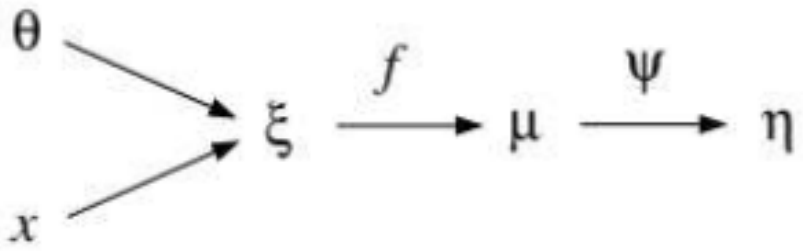
- For using optimization methods that require the Hessian, we can see that:

$$\frac{d\ell_i}{d\theta_j} = (y_i - \mu_i)\, x_{ij} \Rightarrow \frac{d^2\ell_i}{d\theta_k d\theta_j} = -\frac{1}{\sigma^2}\frac{d\mu_i}{d\xi_i}\frac{d\xi_i}{d\theta_k}x_{ij} = -\frac{1}{\sigma^2}\frac{d\mu_i}{d\xi_i}x_{ik}x_{ij}$$

- Thus the Hessian for canonical link is:

$$\boldsymbol{H} = -\frac{1}{\sigma^2}\sum_{i=1}^{N}\frac{d\mu_i}{d\xi_i}\boldsymbol{x}_i\boldsymbol{x}_i^T = -\frac{1}{\sigma^2}\boldsymbol{X}^T\boldsymbol{S}\boldsymbol{X},\ \boldsymbol{S} = diag\left(\frac{d\mu_1}{d\xi_1},...,\frac{d\mu_N}{d\xi_N}\right)$$

- $\boldsymbol{H}$ can be used within an IRLS algorithm where the least squares update is:



$$\boldsymbol{\theta}_{t+1} = (\boldsymbol{X}^T\boldsymbol{S}_t\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{S}_t\boldsymbol{z}_t$$

$$\boldsymbol{z}_t = \boldsymbol{\eta}_t + \boldsymbol{S}_t^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_t) \quad where \quad \boldsymbol{\eta}_t = \boldsymbol{X}\boldsymbol{\theta}_t, \boldsymbol{\mu}_t = f(\boldsymbol{\xi}_t)$$

- For non-canonical links, $\boldsymbol{H}$ has another term. However, the expected $\boldsymbol{H}$ remains is given as above. Using the expected Hessian (Fisher information matrix) instead of the actual $\boldsymbol{H}$ is known as the ***Fisher scoring method***.

- Finally note that it is easy to modify the above procedure to perform MAP estimation with a Gaussian prior.

Bayesian Inference for GLMs

- Bayesian inference for GLMs is usually conducted using MCMC. Possible methods include
  - Metropolis Hastings with an IRLS-based proposal (Gamerman 1997),
  - Gibbs sampling using adaptive rejection sampling (ARS) for each full-conditional (Dellaportas and Smith 1993)
  - Dey et al. provides a review of methodologies.
- It is also possible to use the Gaussian (Laplace) approximation or variational inference.

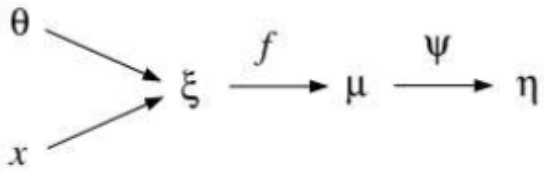
- Gamerman, D. (1997). Efficient sampling from the posterior distribution in generalized linear mixed models. *Statistics and Computing 7*, 57–68.
- Dellaportas, P. and A. F. M. Smith (1993). Bayesian Inference for Generalized Linear and Proportional Hazards Models via Gibbs Sampling. *J. of the Royal Statistical Society. Series C (Applied Statistics) 42*(3), 443–459.
- Dey, D., S. Ghosh, and B. Mallick (Eds.) (2000). *Generalized Linear Models: A Bayesian Perspective*. Chapman & Hall/CRC Biostatistics Series.

Probit Regression

- In binary logistic regression, we use a model of the form $p(y = 1|\boldsymbol{x}_i, \boldsymbol{\theta}) = sigm(\boldsymbol{\theta}^T \boldsymbol{x}_i)$.

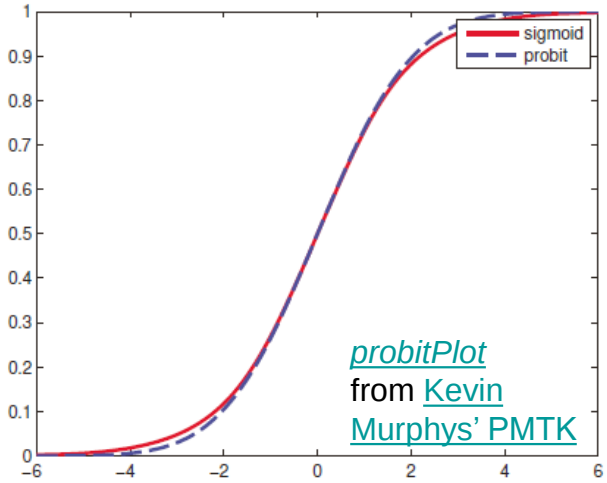
- We can generalize to $p(y = 1|\boldsymbol{x}_i, \boldsymbol{\theta}) = f(\boldsymbol{\theta}^T \boldsymbol{x}_i)$, for any function f that maps $[-\infty, \infty]$ to $[0, 1]$. Possible mean functions are shown here.

Name	Formula
Logistic	$f(\xi) = sigm(\xi) = \frac{e^{\xi}}{1+e^{\xi}}$
Probit	$f(\xi) = \Phi(\xi)$
Log-log	$f(\xi) = exp(-exp(-\xi))$
Complementary log-log	$f(\xi) = 1 - exp(-exp(\xi))$



- We focus now on the case where $f(\xi) = \Phi(\xi)$, where $\Phi(\xi)$ is the cdf of the standard normal.
  This is called *probit regression*.

- The probit function even though similar to the logistic function, it has several advantages over logistic regression.



probitPlot from Kevin Murphys' PMTK

Probit Regression

- We can find the MLE for probit regression using standard gradient methods. Let

$$\mu_i = \boldsymbol{\theta}^T \boldsymbol{x}_i,\ \tilde{y}_i \in \{-1,1\}$$

- Then *the gradient of the log-likelihood for this specific case* is given by

$$\boldsymbol{g}_i = \frac{d}{d\boldsymbol{\theta}} log p(\tilde{y}_i|\boldsymbol{\theta}^T \boldsymbol{x}_i) = \frac{d\mu_i}{d\boldsymbol{\theta}} \frac{d}{d\mu_i} log \Phi(\tilde{y}_i \mu_i) = \boldsymbol{x}_i \frac{\tilde{y}_i \phi(\mu_i)}{\Phi(\tilde{y}_i \mu_i)}$$

- Here ϕ is the standard normal pdf, and Φ is its cdf.

- Similarly, *the Hessian for a single case* is given by

$$\boldsymbol{H}_i = \frac{d^2}{d\boldsymbol{\theta}^2} log p(\tilde{y}_i|\boldsymbol{\theta}^T \boldsymbol{x}_i) = -\boldsymbol{x}_i \left(\frac{\phi(\mu_i)^2}{\Phi(\tilde{y}_i \mu_i)^2} + \frac{\tilde{y}_i \mu_i \ \phi(\mu_i)}{\Phi(\tilde{y}_i \mu_i)} \right) \boldsymbol{x}_i^T$$

- We can also compute the MAP estimate. If we use the prior $p(\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, \boldsymbol{V}_0)$, *the gradient and Hessian of the penalized log likelihood* have the form

$$Gradient: \sum_i \boldsymbol{g}_i + 2\boldsymbol{V}_0^{-1}\boldsymbol{\theta},\ Hessian: \sum_i \boldsymbol{H}_i + 2\boldsymbol{V}_0^{-1}$$

probitRegDemo from Kevin Murphys' PMTK

- These expressions can be used with any gradient-based optimizer.

Interpretation with Latent Variables

- We interpret the probit model as follows: We associate each $\boldsymbol{x}_i$ with two latent utilities, u_{0i} and u_{1i}, corresponding to the possible choices $y_i = 0$ & $y_i = 1$. The observed choice is whichever action has larger utility.

- The model is as follows:

$$u_{oi} = \boldsymbol{\theta}_0^T \boldsymbol{x}_i + \delta_{oi}, u_{1i} = \boldsymbol{\theta}_1^T \boldsymbol{x}_i + \delta_{1i}, y_i = \mathbb{I}\ u_{1i} > u_{oi}$$

- δ's are errors, representing all other factors that we have chosen not to (unable to) model.

- This is called a ***random utility model*** *or* ***RUM****.*

- Since it is only the difference in utilities that matters, let us define $z_i = u_{1i} - u_{0i}$, where $\varepsilon_i = \delta_{1i} - \delta_{0i}$. If the δ's have a Gaussian distribution, then so does ε_i.
  - McFadden, D. (1974). Conditional logit analysis of qualitative choice behavior. In P. Zarembka (Ed.), *Frontiers in econometrics*, pp. 105–142. Academic Press.
  - Train, K. (2009). *Discrete choice methods with simulation*. Cambridge University Press. Second edition

Interpretation with Latent Variables

- Thus we can write

$$z_i = \boldsymbol{\theta}^T \mathbf{x}_i + \varepsilon_i, \boldsymbol{\theta} = \boldsymbol{\theta}_1 - \boldsymbol{\theta}_0, \varepsilon_i = \mathcal{N}(0,1), y_i = 1 = \mathbb{I}\ z_i \geq 0$$

- This is called *the difference RUM or **dRUM** model*.

- When we marginalize out z_i, we can easily see that we recover the probit model:

$$p\ y_i \mid \mathbf{x}_i, \boldsymbol{\theta} = \int \mathbb{I}\ z_i \geq 0\ \mathcal{N}(z_i \mid \boldsymbol{\theta}^T \mathbf{x}_i, 1) dz_i = p\ \boldsymbol{\theta}^T \mathbf{x}_i + \varepsilon \geq 0 = p\ \varepsilon \geq -\boldsymbol{\theta}^T \mathbf{x}_i$$
$$= 1 - \Phi\ -\boldsymbol{\theta}^T \mathbf{x}_i = \Phi\ \boldsymbol{\theta}^T \mathbf{x}_i$$

- Fruhwirth-Schnatter, S. and R. Fruhwirth (2010). Data Augmentation and MCMC for Binary and Multinomial Logit Models. In T. Kneib and G. Tutz (Eds.), *Statistical Modelling and Regression Structures*, pp. 111–132. Springer.

Orbital Probit Regression

- An advantage of the latent variable interpretation of probit regression is that it is easy to extend when the response variable is ordinal, i.e. when it can take on C discrete ordered values, such as low, medium and high. This is called *ordinal regression*.
- The basic idea is as follows. We introduce $C + 1$ thresholds γ_j and set $y_i = j$ if $\gamma_j - 1 < z_i \leq \gamma_j$, where $\gamma_0 \leq \cdots \leq \gamma_C$.
- For identifiability reasons, we set $\gamma_0 = -\infty, \gamma_1 = 0$ and $\gamma_C = \infty$. For example, if $C = 2$, this reduces to the standard binary probit model, whereby $z_i < 0$ produces $y_i = 0$ and $z_i \geq 0$ produces $y_i = 1$.
- If $C = 3$, we partition the real line into 3 intervals: $(-\infty, 0], (0, \gamma_2], (\gamma_2, \infty)$. We can vary γ_2 to ensure the right relative amount of probability mass falls in each interval, so as to match the empirical frequencies of each class label.
- Finding the MLEs for this model is a bit trickier than for binary probit regression, since we *need to optimize for $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$*, and the latter must obey an ordering constraint. EM and Gibbs sampling have been used.

- Kawakatsu, H. and A. Largey (2009). EM algorithms for ordered probit models with endogenous regressors. *The Econometrics Journal 12*(1), 164–186.
- Hoff, P. D. (2009, July). *A First Course in Bayesian Statistical Methods*. Springer.

Multinomial Probit Regression

- Consider a response variable that can take C unordered categorical values, $y_i \in \{1, \ldots, C\}$. The ***multinomial probit*** model is defined as follows:

$$z_{ic} = \boldsymbol{\theta}^T \boldsymbol{x}_{ic} + \varepsilon_{ic}, \;\; \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R}), \; y_i = \arg\max_c z_{ic}$$

- This model is related to ***multinomial logistic regression***. (By defining $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_C]$, and $\boldsymbol{x}_{ic} = [\mathbf{0}, \ldots, \mathbf{0}, \boldsymbol{x}_i, \mathbf{0}, \ldots, \mathbf{0}]$, we can recover the more familiar formulation

$$z_{ic} = \boldsymbol{x}_i^T \boldsymbol{\theta}_c$$

- Since only relative utilities matter, we constrain $\boldsymbol{R}$ to be a correlation matrix.

- If instead of setting $y_i = argmax_c \, z_{ic}$ we use $y_{ic} = \mathbb{I}(z_{ic} > 0)$, we get a model known as ***multivariate probit***, *which is a way to model C correlated binary outcomes.*

- Fruhwirth-Schnatter, S. and R. Fruhwirth (2010). Data Augmentation and MCMC for Binary and Multinomial Logit Models. In T. Kneib and G. Tutz (Eds.), *Statistical Modelling and Regression Structures*, pp. 111–132. Springer.
- Dow, J. and J. Endersby (2004). Multinomial probit and multinomial logit: a comparison of choice models for voting research. *Electoral Studies 23*(1), 107–122
- Talhouk, A., K. Murphy, and A. Doucet (2011). Efficient Bayesian Inference for Multivariate Probit Models with Sparse Inverse Correlation Matrices. *J. Comp. Graph. Statist.*.

Multitask Learning

- Often we want to fit many related classification or regression models. It is reasonable to assume the input-output mapping is similar across these different models, so we can get *better performance by fitting all the parameters at the same time.*
- This is called **multi-task learning** and is tackled using hierarchical Bayesian models.
- Let y_{ij} be the response of the i'th item in group j, for $i = 1:N_j$ and $j = 1:J$.
  - For example, j might index schools, i might index students within a school, and y_{ij} might be the test score.
  - Or j might index people, and i might index purchases, and y_{ij} might be the identity of the item that was purchased (this is known as **discrete choice modeling**).

- Caruana, R. (1998). A dozen tricks with multitask learning. In G. Orr and K.-R. Mueller (Eds.), *Neural Networks: Tricks of the Trade*. Springer-Verlag.
- Raina, R., A. Ng, and D. Koller (2005). Transfer learning by constructing informative priors. In *NIPS*.
- Bakker, B. and T. Heskes (2003). Task Clustering and Gating for Bayesian Multitask Learning. *J. of Machine Learning Research 4*, 83–99.
- Chai, K. M. A. (2010). *Multi-task learning with Gaussian processes*. Ph.D. thesis, U. Edinburgh.

Multitask Learning

- Let $\boldsymbol{x}_{ij}$ be a feature vector associated with y_{ij}. The goal is to fit the models $p(y_j | \boldsymbol{x}_j)$ for all j.
- Some groups have lots of data but the majority of groups have little data. We can use the same model for all groups or fit a separate model for each group but take the model parameters to be similar across groups.
- More precisely, suppose $\mathbb{E}\left[y_{ij} | \boldsymbol{x}_{ij}\right] = f\left(\boldsymbol{x}_{ij}^T \boldsymbol{\beta}_j\right)$ where f is a link function.
- Also let $\boldsymbol{\beta}_j \sim \mathcal{N}\left(\beta_*, \sigma_j^2 \boldsymbol{I}\right), \beta_* \sim \mathcal{N}\left(\boldsymbol{\mu}, \sigma_*^2 \boldsymbol{I}\right)$
- In this model, *groups with small sample size borrow statistical strength from the groups with larger sample size, because the $\boldsymbol{\beta}_j$'s are correlated via the latent common parents $\boldsymbol{\beta}_*$.*
- The term σ_j^2 controls how much group *j* depends on the common parents and the σ_*^2 term controls the strength of the overall prior.
- Suppose, for simplicity, that $\boldsymbol{\mu} = \mathbf{0}$, and that σ_j^2, σ_*^2 are all known (e.g., they could be set by cross validation).

Multitask Learning

- The log probability has the form:

$$\log p(\mathcal{D} \mid \beta) + \log p(\beta) = \sum_j \left[\log p\left(\mathcal{D}_j \mid \beta_j\right) - \frac{\left\|\beta_j - \beta_*\right\|^2}{2\sigma_j^2} \right] - \frac{\left\|\beta_*\right\|^2}{2\sigma_*^2}$$

- We can perform MAP estimation of $\beta = \beta_{1:J}, \beta_*$ using gradient methods.
- One can also alternate between optimizing the β_j and the β_*. Since the likelihood and the prior are convex this will converge to the global optimum.
- Once the model is trained, we discard β_* and use each model separately.

Multitask Learning: Personalized Spam Filtering

- We want to fit one classifier per user, $\boldsymbol{\beta}_j$. Since most users do not label their email as spam or not, it is hard to estimate these models independently. *Let $\boldsymbol{\beta}_j$ have **a common prior $\boldsymbol{\beta}_*$**, representing the parameters of a generic user.*
- In this case, we can emulate the behavior of the above model with *a simple trick: we make two copies of each feature $\boldsymbol{x}_i$, one concatenated with the user id, and one not.* The effect will be to learn a predictor of the form (u =user ID):

$$\mathbb{E}\left[y_i \mid \boldsymbol{x}_i, u\right] = \left(\boldsymbol{\beta}_*, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_J\right)^T \left[\boldsymbol{x}_i, \mathbb{I}(u=1)\boldsymbol{x}_i, \ldots \mathbb{I}\left(u=J\right)\boldsymbol{x}_i\right]$$

$$\mathbb{E}\left[y_i \mid \boldsymbol{x}_i, u=j\right] = \left(\boldsymbol{\beta}_* + \boldsymbol{w}_j\right)^T \boldsymbol{x}_i$$

- Thus $\boldsymbol{\beta}_*$ will be estimated from everyone's email, whereas $\boldsymbol{w}_j$ will just be estimated from the user j's email.
- To see correspondence with the hierarchical Bayesian model, let $\boldsymbol{w}_j = \boldsymbol{\beta}_j - \boldsymbol{\beta}_*$.
- Then the log probability of the original model can be rewritten as

$$\sum_j \left[\log p\left(D_j \mid \boldsymbol{\beta}_* + \boldsymbol{w}_j\right) - \frac{\left\|\boldsymbol{w}_j\right\|^2}{2\sigma_j^2}\right] - \frac{\left\|\boldsymbol{\beta}_*\right\|^2}{2\sigma_*^2}$$

- Daume, H. (2007b). Frustratingly easy domain adaptation. In *Proc. the Assoc. for Comp. Ling.*
- Attenberg, J., K. Weinberger, A. Smola, A. Dasgupta, and M. Zinkevich (2009). Collaborative spam filtering with the hashing trick. In *Virus Bulletin*.
- Weinberger, K., A. Dasgupta, J. Attenberg, J. Langford, and A. Smola (2009). Feature hashing for large scale multitask learning. In *Intl. Conf. on Machine Learning*.

Multitask Learning: Personalized Spam Filtering

$$\sum_j \left[\log p\left(D_j \mid \boldsymbol{\beta}_* + \boldsymbol{w}_j \right) - \frac{\left\| \boldsymbol{w}_j \right\|^2}{2\sigma_j^2} \right] - \frac{\left\| \boldsymbol{\beta}_* \right\|^2}{2\sigma_*^2}$$

- If we assume $\sigma_j^2 = \sigma_*^2$, the effect is the same as using the augmented feature trick, with the same regularizer strength for both $\boldsymbol{w}_j$ and $\boldsymbol{\beta}_*$.
- One gets better performance by not requiring that $\sigma_j^2 = \sigma_*^2$

- Finkel, J. and C. Manning (2009). Hierarchical bayesian domain adaptation. In *Proc. NAACL*, pp. 602–610

Multitask Learning: Domain Adaptation

- This is the problem of training a set of classifiers on data drawn from different distributions (e.g. email & news articles text).
- This problem is a special case of multi-task learning, where *the tasks are the same.*
- The above hierarchical Bayesian model was used to perform domain adaptation for two Natural Language Processing (NLP) tasks, namely named entity recognition and parsing.
- Large improvements over fitting separate models to each dataset, but small improvements when pooling all the data to fit a single model.

- Finkel, J. and C. Manning (2009). Hierarchical bayesian domain adaptation. In *Proc. NAACL*, pp. 602–610

Priors in Multitask Learning

- Other than Gaussian priors are often more suitable.
- Consider *conjoint analysis*: *figuring out which features of a product customers like best.*
- This can be modelled with the hierarchical Bayesian setup, but *using a sparsity-promoting prior on $\boldsymbol{\beta}_j$, rather than a Gaussian prior*.
- This is called *multi-task feature selection*.
- Often we cannot assume that all tasks are all equally similar. If we pool the parameters across tasks that are qualitatively different, the performance will be worse than not using pooling, because the inductive bias of our prior is wrong (*negative transfer*).

- Lenk, P., W. S. DeSarbo, P. Green, and M. Young (1996). Hierarchical Bayes Conjoint Analysis: Recovery of Partworth Heterogeneity from Reduced Experimental Designs. *Marketing Science 15*(2), 173– 191.
- Argyriou, A., T. Evgeniou, and M. Pontil (2008). Convex multi-task feature learning. *Machine Learning 73*(3), 243–272.

Priors in Multitask Learning

- One way around this problem is to *use as prior a mixture of Gaussians*. This provides robustness against prior mis-specification.

- One also can *combine Gaussian mixtures with sparsity-promoting priors*.

- Ji, S., D. Dunson, and L. Carin (2009). Multi-task compressive sensing. *IEEE Trans. Signal Processing 57* (1)
- Xu, F. and J. Tenenbaum (2007). Word learning as Bayesian inference. *Psychological Review 114*(2).
- Jacob, L., F. Bach, and J.-P. Vert (2008). Clustered Multi-Task Learning: a Convex Formulation. In *NIPS*.

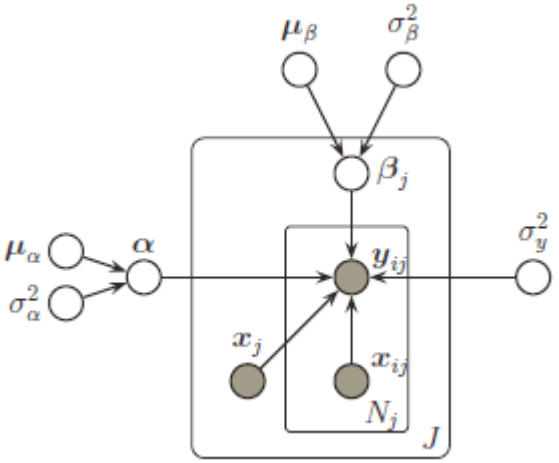
Generalized Linear Mixed Models

- We now generalize multi-task learning to allow the response to *include information at the group level, $\boldsymbol{x}_j$, as well as at the item level, $\boldsymbol{x}_{ij}$.* Similarly, we can allow the *parameters to vary across groups, $\boldsymbol{\beta}_j$* , or *to be tied across groups, $\boldsymbol{\alpha}$*. This leads to the following model:

$$\mathbb{E}\left[y_{ij} \mid \boldsymbol{x}_{ij}, \boldsymbol{x}_j \right] = f\left(\phi_1\left(\boldsymbol{x}_{ij}\right)^T \boldsymbol{\beta}_j + \phi_2\left(\boldsymbol{x}_j\right)^T \boldsymbol{\beta}_j' + \phi_3\left(\boldsymbol{x}_{ij}\right)^T \boldsymbol{\alpha} + \phi_4\left(\boldsymbol{x}_j\right)^T \boldsymbol{\alpha}' \right)$$

- Here ϕ_k are basis functions. Note that the number of $\boldsymbol{\beta}_j$ parameters grows with the number of groups, whereas the size of $\boldsymbol{\alpha}$ is fixed.

- The terms β_j *are called* random effects (they vary randomly across groups), and $\boldsymbol{\alpha}$ *are called the* fixed effect (fixed but unknown constant).

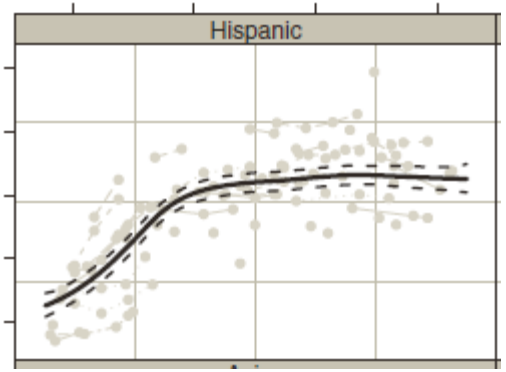


- A model with both fixed and random effects is called a *mixed model*. If $p(y|\boldsymbol{x})$ is a GLM, the overall model is called a ***generalized linear mixed effects model or GLMM*****.**

- Wand, M. (2009). Semiparametric regression and graphical models. *Aust. N. Z. J. Stat. 51*(1), 9–41.

Generalized Linear Mixed Models

- Consider the following example. Suppose y_{ij} is the amount of spinal bone mineral density (SBMD) for person j at measurement i. Let x_{ij} be the age of person, and let x_j be their ethnicity: White, Asian, Black, or Hispanic.
- The primary goal is to determine if there are significant differences in the mean SBMD among the four ethnic groups accounting for age.
- There is a nonlinear effect of SBMD vs age, so we use a *semi-parametric model which combines linear regression with non-parametric regression.*



- We also see that *there is variation across individuals within each group, so we will use a mixed effects model*. Specifically, we will use $\boldsymbol{\phi}_1(\boldsymbol{x}_{ij}) = 1$ to account for the random effect of each person; $\boldsymbol{\phi}_2(\boldsymbol{x}_{ij}) = 0$ since no other coefficients are person-specific; $\boldsymbol{\phi}_3(\boldsymbol{x}_{ij}) = [b_k(x_{ij})]$, where b_k is the k'th spline basis functions, to account for the nonlinear effect of age; and

$$\phi_4(x_j) = \left(\mathbb{I}(x_j = w), \mathbb{I}(x_j = a), \mathbb{I}(x_j = b), \mathbb{I}(x_j = h)\right)$$

to account for the effect of the different ethnicities.

- Wand, M. (2009). Semiparametric regression and graphical models. *Aust. N. Z. J. Stat. 51*(1), 9–41.
- Ruppert, D., M. Wand, and R. Carroll (2003). *Semiparametric Regression*. Cambridge University Press

Generalized Linear Mixed Models

- The overall model is therefore

$$\mathbb{E}\left[y_{ij} \mid x_{ij}\ x_j \right] = \beta_j + \boldsymbol{\alpha}^T \mathbf{b}\left(x_{ij}\right) + \underbrace{\varepsilon_{ij}}_{\sim \mathcal{N}\left(0,\sigma_y^2\right)} + \alpha_w' \mathbb{I}\left(x_j = w\right) + \alpha_a' \mathbb{I}\left(x_j = a\right) + \alpha_b' \mathbb{I}\left(x_j = b\right) + \alpha_h' \mathbb{I}\left(x_j = h\right)$$

- $\boldsymbol{\alpha}$ contains the non-parametric part of the model related to age, $\boldsymbol{\alpha}'$ *contains the parametric part of the model related to ethnicity*, and β_j *is a random offset for person j.*

- We endow all of these regression coefficients with separate Gaussian priors.

- One can perform posterior inference to compute $p(\boldsymbol{\alpha}, \boldsymbol{\alpha}', \boldsymbol{\beta}, \sigma^2 | \mathcal{D})$

- After fitting, we can compute the prediction for each group.

- We can also perform significance testing, by computing $p(\alpha_g - \alpha_w | \mathcal{D})$ for each ethnic group g relative to some baseline (e.g. white).

- Wand, M. (2009). Semiparametric regression and graphical models. *Aust. N. Z. J. Stat. 51*(1), 9–41.
- Ruppert, D., M. Wand, and R. Carroll (2003). *Semiparametric Regression*. Cambridge University Press

Computational Issues

- GLMMs are difficult to fit. At first, $p(y_{ij}|\boldsymbol{\theta})$ may not be conjugate to the prior $p(\boldsymbol{\theta})$ where $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\beta})$.
- Secondly, the regression coefficients $\boldsymbol{\theta}$ and the means and variances of the priors $\boldsymbol{\eta} = (\boldsymbol{\mu}, \boldsymbol{\sigma})$ are unknown.
- One approach is to use fully Bayesian inference methods, such as *variational Bayes* or *MCMC*.
- An alternative approach is to use *empirical Bayes*.
- In the context of a GLMM, we can use *the EM algorithm*, where in the E step we compute $p(\boldsymbol{\theta}|\boldsymbol{\eta}, \mathcal{D})$, and in the M step we optimize $\boldsymbol{\eta}$.
- In general for the E step we need to use approximations (numerical quadrature or Monte Carlo).

- Hall, P., J. T. Ormerod, and M. P. Wand (2011). Theory of Gaussian Variational Approximation for a Generalised Linear Mixed Model. *Statistica Sinica 21*, 269–389.
- Gelman, A. and J. Hill (2007). *Data analysis using regression and multilevel/hierarchical models*. Cambridge.
- Breslow, N. E. and D. G. Clayton (1993). Approximate inference in generalized linear mixed models. *J. of the Am. Stat. Assoc. 88*(421), 9–25.

Computational Issues

- A faster approach is to use *variational EM*.
- In (frequentist) statistics, there is a popular method for fitting GLMMs called *generalized estimating equations or GEE.*
- This approach is not recommended as it is not as statistically efficient as likelihood-based methods.
- In addition, it can only provide estimates of the population parameters $\boldsymbol{\alpha}$, but not the random effects $\boldsymbol{\beta}_j$.

- Braun, M. and J. McAuliffe (2010). Variational Inference for Large-Scale Models of Discrete Choice. *J. of the Am. Stat. Assoc. 105*(489), 324–335.
- Hardin, J. and J. Hilbe (2003). *Generalized Estimating Equations*. Chapman and Hall/CRC.

Learning to Rank (LETOR)

- We want to learn a function that can rank order a set of items (e.g. in information retrieval).
- Specifically, suppose we have a query q and a set of documents $d_1, \ldots, d_m$ that might be relevant to q (e.g., all documents that contain the string q).
- We would like to sort these documents in decreasing order of relevance and show the top k to the user. Similar problems arise in collaborative filtering.
- A standard way to measure the relevance of a document d to a query q is to use a probabilistic language model based on a bag of words' model. That is, we define

$$sim(q,d) = p(q \mid d) = \prod_{i=1}^{n} p(q_i \mid d)$$

where q_i is the i'th word or term, and $p(q_i|d)$ is a multinoulli distribution estimated from document d.

- Liu, T.-Y. (2009). Learning to rank for information retrieval. *Found. & Trends in Inform. Retrieval 3*(3), 225–331.

Learning to Rank (LETOR)

- In practice, we need to smooth the estimated distribution, e.g. using a *Dirichlet prior, representing the overall frequency of each word*. This can be estimated from all documents in the system:

$$p(t \mid d) = (1-\lambda)\frac{TF(t,d)}{LEN(d)} + \lambda\, p(t \mid background)$$

- Here $TF(t,d)$ is the frequency of term t in document d, $LEN(d)$ is the number of words in d, and $0 < \lambda < 1$ is a smoothing parameter.

- Liu, T.-Y. (2009). Learning to rank for information retrieval. *Found. & Trends in Inform. Retrieval 3*(3), 225–331.

Learning to Rank

$$p(t \mid d) = (1-\lambda)\frac{TF(t,d)}{LEN(d)} + \lambda\, p(t \mid background)$$

- Here $TF(t,d)$ is the frequency of term t in document d, $LEN(d)$ is the number of words in d, and $0 < \lambda < 1$ is a smoothing parameter.
- There are many other signals that we can use to measure relevance.
- For example, the *PageRank of a web document is a measure of its authoritativeness*, derived from the web's link structure.
- We can also compute how often and where the query occurs in the document.

- Liu, T.-Y. (2009). Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval 3*(3), 225–331.
- Zhai, C. and J. Lafferty (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Trans. on Information Systems 22*(2), 179–214.

Learning to Rank: Pointwise Approach

- Suppose we collect some training data representing the relevance of a set of documents for each query.
- Specifically, for *each query* q, suppose that we retrieve m possibly relevant documents d_j, for $j = 1:m$.
- For each query document pair *define a feature vector,* $\boldsymbol{x}(q,d)$. This contains e.g. the query-document similarity score & page rank score of the document.
- Suppose we have a set of labels y_j representing the degree of relevance of document d_j to query q. Such labels might be binary (relevant/irrelevant), or may represent a degree of relevance (very relevant/somewhat relevant/ irrelevant).
- Such labels can be obtained from query logs, by thresholding the number of times a document was clicked on for a given query.
- If we have binary relevance labels, we can solve the problem using a standard binary classification scheme to estimate, $p(y = 1|\boldsymbol{x}(q,d))$.

Learning to Rank: Pointwise Approach

- *If we have ordered relevancy labels*, we can use *ordinal regression* to predict the rating, $p(y = r|\boldsymbol{x}(q,d))$.

- In both binary and ordered relevancy label cases, we can then sort the documents by this scoring metric. This is called the *pointwise approach* to Learning to Rank (LETOR), and is widely used because of its simplicity.

- This method does not take into account the location of each document in the list. Thus it penalizes errors at the end of the list just as much as errors at the beginning. In addition, each decision about relevance is made very myopically.

Learning to Rank: Pairwise Approach

- People are better at judging the relative relevance of two items rather than absolute relevance.
- Consequently, the data might tell us that d_j is more relevant than d_k for a given query, or vice versa.
- We can model this kind of data using a binary classifier of the form

$$p\left(y_{jk} \mid \boldsymbol{x}(q,d_j), \boldsymbol{x}(q,d_k)\right)$$

- *Here we* set $y_{jk} = 1$ if $rel(d_j,q) > rel(d_k,q)$ and $y_{jk} = 0$ otherwise.
- One way to model such a function is as follows:

$$p\left(y_{jk}=1, \boldsymbol{x}_j, \boldsymbol{x}_k\right) = sigm\left(f\left(\boldsymbol{x}_j\right) - f\left(\boldsymbol{x}_k\right)\right)$$

- $f(\boldsymbol{x})$ is a scoring function, e.g. $f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$.

- Carterette, B., P. Bennett, D. Chickering, and S. Dumais (2008). Here or There: Preference Judgments for Relevance. In *Proc. ECIR*

Learning to Rank: Pairwise Approach

$$p\left(y_{jk}=1,\boldsymbol{x}_j,\boldsymbol{x}_k\right)=sigm\left(f\left(\boldsymbol{x}_j\right)-f\left(\boldsymbol{x}_k\right)\right)$$

- This is a special kind of neural network known as *RankNet.*
- We can find the MLE of $\boldsymbol{w}$ by maximizing the log likelihood, or equivalently, by *minimizing the cross entropy loss*, given by

$$L=\sum_{i=1}^{N}\sum_{j=1}^{m_i}\sum_{k=j+1}^{m_i}L_{ijk}$$

$$-L_{ijk}=\mathbb{I}\left(y_{ijk}=1\right)\log p\left(y_{ijk}=1\mid\boldsymbol{x}_{ij},\boldsymbol{x}_{ik},\boldsymbol{w}\right)$$

$$+\mathbb{I}\left(y_{ijk}=0\right)\log p\left(y_{ijk}=0\mid\boldsymbol{x}_{ij},\boldsymbol{x}_{ik},\boldsymbol{w}\right)$$

- This can be optimized using gradient descent.
- A variant of $RankNet$ is used by Microsoft's Bing search engine.

- Carterette, B., P. Bennett, D. Chickering, and S. Dumais (2008). Here or There: Preference Judgments for Relevance. In *Proc. ECIR*
- Burges, C. J., T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender (2005). Learning to rank using gradient descent. In *Intl. Conf. on Machine Learning*, pp. 89–96.

Learning to Rank: The Listwise Approach

- In pairwise approach decisions about relevance are made just based on a pair of items (documents), rather than considering the full context.
- Consider methods that look at the entire list of items at the same time.
- We can define a total order on a list by specifying a permutation of its indices, $\boldsymbol{\pi}$. To model our uncertainty about $\boldsymbol{\pi}$, we can use the *Plackett-Luce distribution*. This has the following form:

$$p(\boldsymbol{\pi} \mid \boldsymbol{s}) = \prod_{j=1}^{m} \frac{s_j}{\sum_{u=j}^{m} s_u}$$

- $s_j = s(\pi^{-1}(j))$ is the score of the document ranked at the j'th position.
- As an example, suppose $\boldsymbol{\pi} = (A, B, C)$. Then we have that $p(\boldsymbol{\pi})$ is the probability of A being ranked first, times the probability of B being ranked second given that A is ranked first, times the probability of C being ranked third given that A and B are ranked first and second. In other words,

$$p(\boldsymbol{\pi} \mid \boldsymbol{s}) = \frac{s_A}{s_A + s_B + s_C} \times \frac{s_B}{s_B + s_C} \times \frac{s_C}{s_C}$$

- Plackett, R. (1975). The analysis of permutations. *Applied Stat. 24*, 193–202.
- Luce, R. (1959). *Individual choice behavior: A theoretical analysis*. Wiley

Learning to Rank: The Listwise Approach

- To incorporate features, we can define $s(d) = f(\boldsymbol{x}(q,d))$, where we often take f to be a linear function, $f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$. This is known as the *ListNet model*.

- To train this model, let $\boldsymbol{y}_i$ be the relevance scores of the documents for query i. We then minimize the cross entropy term

$$-\sum_i \sum_\pi p(\pi \mid y_i) \log p(\pi \mid s_i)$$

- This is intractable, since the i'th term needs to sum over $m_i!$ permutations. To make it tractable, we consider permutations over the top k positions only:

$$p\left(\pi_{1:k} \mid s_{1:m}\right) = \prod_{j=1}^{k} \frac{s_j}{\sum_{u=j}^{m} s_u}$$

- Cao, Z., T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li (2007). Learning to rank: From pairwise approach to listwise approach. In *Intl. Conf. on Machine Learning*, pp. 129a ˇA, S136.
- Yang, S., B. Long, A. Smola, H. Zha, and Z. Zheng (2011). Collaborative competitive filtering: learning recommender using context of user choice. In *Proc. Annual Intl. ACM SIGIR Conference*.

Learning to Rank: The Listwise Approach

- There are only $m!/(m - k)!$ such permutations. If we set $k = 1$, we can evaluate each cross entropy term (and its derivative) in $\mathcal{O}(m)$ time.
- In the special case when only one document from the presented list is deemed relevant, say $y_i = c$, we can use multinomial logistic regression:

$$p(y_i = c \mid \boldsymbol{x}) = \frac{\exp(s_c)}{\sum_{c'} \exp(s_{c'})}$$

- This performs as well as ranking methods for collaborative filtering.

- Cao, Z., T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li (2007). Learning to rank: From pairwise approach to listwise approach. In *Intl. Conf. on Machine Learning*, pp. 129a ˇA, S136.
- Yang, S., B. Long, A. Smola, H. Zha, and Z. Zheng (2011). Collaborative competitive filtering: learning recommender using context of user choice. In *Proc. Annual Intl. ACM SIGIR Conference*.

Loss Functions for Ranking

- There are various ways to measure the performance of a ranking system.

- *Mean reciprocal rank (MRR).* For a query q, let the rank position of its first relevant document be denoted by $r(q)$. Then we define the mean reciprocal rank to be $1/r(q)$. This is a very simple performance measure.

- *Mean average precision (MAP).* In the case of binary relevance labels, we can define the precision at k of some ordering as follows:

$$p@k(\pi) = \frac{num.\ relevant\ documents\ in\ the\ top\ k\ \ positions\ of\ \ \pi}{k}$$

We then define the *average precision* as follows:

$$AP(\pi) = \frac{\sum_k P@k(\pi) \cdot I_k}{num.\ relevant\ documents}$$

Here I_k is 1 iff document *k* is relevant. For example, if we have the relevancy labels $\boldsymbol{y} = (1, 0, 1, 0, 1)$, then the AP is $1/3\ (\ 1/1\ +\ 2/3\ +\ 3/5\ )\ \approx\ 0.76$.

Finally, we define the *mean average precision* as the AP averaged over all queries.

Normalized Discounted Cumulative Gain

- *Normalized discounted cumulative gain (NDCG).* Suppose the relevance labels have multiple levels. We can define the discounted cumulative gain of the first k items in an ordering as follows:

$$DCG@k(r) = r_1 + \sum_{i=2}^{k} \frac{r_i}{\log_2 i}$$

where r_i is the relevance of item i and the log_2 term is used to discount items later in the list.

An alternative definition, that places stronger emphasis on retrieving relevant documents, uses

$$DCG@k(r) = \sum_{i=1}^{k} \frac{2^{r_i} - 1}{\log_2(1+i)}$$

The trouble with DCG is that it varies in magnitude just because the length of a returned list may vary. It is common to normalize this measure by the ideal DCG, i.e. one obtained by optimal ordering: $IDCG@k(\boldsymbol{r}) = argmax_{\boldsymbol{\pi}} DCG@k(\boldsymbol{r})$

This can be easily computed by sorting $r_{1:m}$ and then computing $DCG@k$. Finally, we define the *normalized discounted cumulative gain or NDCG* as $DCG/IDCG$. The $NDCG$ can be averaged over queries to give a measure of performance.

Rank Correlation

- *Rank correlation*. We can measure the correlation between the ranked list, $\boldsymbol{\pi}$, and the relevance judgement, $\boldsymbol{\pi}^*$, using a variety of methods.

- One approach, known as the (weighted) Kendall's τ statistics, is defined in terms of the weighted pairwise inconsistency between the two lists:

$$\tau(\pi, \pi^*) = \frac{\sum_{u<v} w_{uv}\left[1+\text{sgn}\left(\pi_u - \pi_v\right)\text{sgn}\left(\pi_u^* - \pi_v^*\right)\right]}{2\sum_{u<v} w_{uv}}$$

Loss Functions for Ranking

- These loss functions can be used in different ways.
- In the Bayesian approach, we first fit the model using posterior inference; this depends on the likelihood and prior, but not the loss.
- We then choose our actions at test time to minimize the expected future loss.
- One way to do this is to sample parameters from the posterior, $\boldsymbol{\theta}^s \sim p(\boldsymbol{\theta}|D)$, and then evaluate, say, the $precision@k$ for different thresholds, averaging over $\boldsymbol{\theta}^s$

- Zhang, X., T. Graepel, and R. Herbrich (2010). Bayesian Online Learning for Multi-label and Multi-variate Performance Measures. In *AI/Statistics*..

Loss Functions for Ranking

- In the frequentist approach, we try to minimize the empirical loss on the training set. The problem is that these loss functions are not differentiable functions of the model parameters.

- We can either use gradient-free optimization methods, or we can minimize a surrogate loss function instead. Cross entropy loss (i.e., negative log likelihood) is an example of a widely used surrogate loss function.

- Another loss, known as *weighted approximate-rank pairwise or* $WARP$ loss provides a better approximation to the $precision@k$ loss. $WARP$ is defined as follows:

$$WARP\left(f\left(\mathbf{x},:\right),y\right)=L\left(rank\left(f\left(\mathbf{x},:\right),y\right)\right)$$

$$rank\left(f\left(\mathbf{x},:\right),y\right)=\sum_{y\neq y'}\mathbb{I}\left(f\left(\mathbf{x},y'\right)\geq f\left(\mathbf{x},y'\right)\right)$$

$$L\left(k\right)=\sum_{j=1}^{k}\alpha_j,\ \alpha_1\geq\alpha_2\geq\ldots\geq 0$$

  - Usunier, N., D. Buffoni, and P. Gallinari (2009). Ranking with ordered weighted pairwise classification.

Loss Functions for Ranking

- Here $\boldsymbol{f}\ (x, :)\ =\ [f(\boldsymbol{x}, 1), \ldots, f(\boldsymbol{x}, |y|)]$ is the vector of scores for each possible output label, or, in IR terms, for each possible document corresponding to input query $\boldsymbol{x}$.

- The expression $rank(\boldsymbol{f}\ (\boldsymbol{x}, :), y)$ measures the rank of the true label y assigned by this scoring function.

- Finally, L transforms the integer rank into a real-valued penalty. Using $\alpha_1\ =\ 1$ and $\alpha_{j>1}\ =\ 0$ would optimize the proportion of top-ranked correct labels.

- Setting $\alpha_{1.\,k}$ to be non-zero values would optimize the top k in the ranked list, which will induce good performance as measured by MAP or $precision@k$.

- As it stands, $WARP$ loss is still hard to optimize, but it can be further approximated by Monte Carlo sampling, and then optimized by gradient descent.

- Weston, J., S. Bengio, and N. Usunier (2010). Large Scale Image Annotation: Learning to Rank with Joint Word-Image Embeddings. In *Proc. European Conf. on Machine Learning*